

Büther, Marcel

**Working Paper**

## Reducing the elastic generalized assignment problem to the standard generalized assignment problem

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 632

**Provided in Cooperation with:**

Christian-Albrechts-University of Kiel, Institute of Business Administration

*Suggested Citation:* Büther, Marcel (2007) : Reducing the elastic generalized assignment problem to the standard generalized assignment problem, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 632, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/147551>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

Manuskripte  
aus den  
Instituten für Betriebswirtschaftslehre  
der Universität Kiel

No. 632

**Reducing the Elastic Generalized Assignment Problem  
to the Standard Generalized Assignment Problem**

Marcel Büther

Dezember 2007

Marcel Büther  
Christian-Albrechts-Universität zu Kiel,  
Institut für Betriebswirtschaftslehre,  
Olshausenstr. 40, 24098 Kiel, Germany,  
<http://www.bwl.uni-kiel.de/bwlinstitute/Prod>  
[buether@bwl.uni-kiel.de](mailto:buether@bwl.uni-kiel.de)

## Abstract

The elastic generalized assignment problem (eGAP) is a natural extension of the generalized assignment problem (GAP) where the capacities are not fixed but can be adjusted which is expressed by continuous variables. These variables might be unbounded or restricted by a lower or upper bound, respectively. This paper concerns techniques in order to reduce several variants of eGAP to GAP which enables us to employ standard approaches for the GAP. This results into an heuristic, which can be customized in order to provide solutions having an objective value arbitrarily close to the optimal one.

**Keywords:** elastic generalized assignment problem, mixed integer programming, reformulation, binary representation

## 1 Introduction

The elastic generalized assignment problem, eGAP for short, is a natural extension of the generalized assignment problem (GAP), a well-known combinatorial optimization problem with applications in production, logistics, and distribution planning. The GAP is to assign jobs to agents in order to minimize assignment cost without extending a given capacity for each agent. While the capacity can not be influenced according to the GAP, the eGAP considers the opportunity to extend or reduce, respectively, the available capacity. Extending the capacity causes cost while reducing capacity provides a profit. The GAP is well-known to be NP-hard, NP-hardness of eGAP follows straightforwardly, see Nauss [9].

While there exist several efficient algorithms for the GAP, see, e.g. Yagiura et al. [17], the eGAP has not been studied much so far, although it is of great practical relevance. To the best of our knowledge, there is only one recent paper by Nauss [9] for a special case of the eGAP. It concerns a branch and bound algorithm to solve small but difficult instances. This algorithm is based on a previous work, which was developed for the GAP, see Nauss [8].

Due to the possibility of adjusting the capacity the well-known cover inequalities, see Nemhauser and Wolsey [10], Schrijver [12] as well as Fügenschuh and Martin [3], can not be used one-to-one. Therefore, the powerful formulations developed by Wolsey [14] have to be adapted. This seems to be expensive here, see Ceria et al. [2], due to the number of possible cuts. Thus we propose to reduce the problem to the well-known GAP, for which several efficient algorithms exist.

The exposition of our work is as follows: In section 2 we present the mixed integer formulation and develop some properties of solutions. The approaches with one variable for the adjustment at each agent, being unbounded or limited by one or two bounds, respectively, are given in section 3. Then, we give a generalization by introducing additional variables for the adjustment for each agent in section 4. In section 5 we provide some computational results and section 6 finally concludes the paper.

## 2 Model Formulation and First Insights

Given a set  $M = \{1, \dots, m\}$  of agents and a set  $N = \{1, \dots, n\}$  of jobs, the objective of the eGAP is to serve all jobs at minimum cost, regarding the adjusted capacity of each agent. To

this end, the following binary decision variables are used:

$$x_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to agent } i \\ 0 & \text{otherwise} \end{cases}$$

The binary variables ensure that a job can only be served fully by an agent.

The following parameters are assumed to be given: Each agent  $i$  provides a fixed amount of capacity  $b_i$  ( $b_i > 0$ ) of an arbitrary resource. In contrast to the GAP, the original capacity  $b_i$ , can be adjusted. Continuous variable  $s_i$  represents the amount of capacity which is added ( $s_i > 0$ ) or removed ( $s_i < 0$ ), respectively. The cost for the capacity adjustment are  $c_i$ ,  $c_i > 0$ .

Job  $j$  asks for a specific amount  $a_{ij}$  ( $a_{ij} > 0$ ) of the capacity of a particular agent if  $x_{ij} = 1$ . For serving job  $j$ 's demand from agent  $i$ , the total cost  $c_{ij}$  ( $c_{ij} \geq 0$ ) occur. These cost needn't depend on the value  $a_{ij}$ .

Now, the problem can be stated mathematically as follows:

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{i \in M} c_i s_i \quad (1)$$

$$\text{s.t. } \sum_{j \in N} a_{ij} x_{ij} \leq b_i + s_i \quad \forall i \in M \quad (2)$$

$$\sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \quad (3)$$

$$l_i \leq s_i \leq u_i \quad \forall i \in M \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N \quad (5)$$

The objective function (1) is twofold. The first part represents the sum of assigned jobs' cost. The second part considers cost and sales when additional capacity is bought and superfluous capacity is sold, respectively. Constraints (2) ensure that the (modified) capacity limit of the agents is not exceeded. Constraints (3) in combination with (5) ensure, that each job is assigned exactly to one agent. Finally, constraints (4) give the bounds for the capacity adjustment.

Before presenting reductions of eGAP to GAP for most cases, we introduce some definitions and basic properties, first.

For short, a solution of the eGAP can be expressed by the tuple  $(x, s)$ , where  $x$  is the binary vector containing all values of  $x_{ij}$ ,  $i \in M$  and  $j \in N$ , and  $s$  is the vector of the corresponding capacity modifications  $s_i$ ,  $i \in M$ . Additionally, we define the following capacity consumptions:

$$W_i := \sum_{j \in N} a_{ij} \quad \forall i \in M$$

where  $W_i$  gives the capacity consumption at agent  $i$ , if all jobs are assigned to this agent.

Note, that the number of feasible solution of the eGAP is not smaller than that of a corresponding GAP with capacities of  $b_i + l_i$ , for each  $i \in M$ . This might be a burden, if an heuristic is used to solve the eGAP, which is based on local search, see e.g. Glover et al. [4].

### 3 Capacity Adjustment with one Variable

#### 3.1 Capacity Adjustment with no restrictive Bounds

Before dealing with the one-sided and the two-sided limitation of the capacity adjustment in the following subsections, we first highlight the special case, where  $s_i$  is not limited by any effective bound, e.g.  $l_i \leq -b_i$  and  $u_i \geq W_i - b_i \forall i \in M$ . Obviously, for each solution constraint (2) is tight.

**Lemma 1.** *Given an optimal solution  $(x, s)$  with  $\sum_{j \in N} a_{ij}x_{ij} \geq b_i + l_i$  for any  $i \in M$ , the corresponding constraint (2) is fulfilled with equality for this agent.*

*Proof.* Suppose for any  $i \in M$ ,  $b_i + l_i \leq \sum_{j \in N} a_{ij}x_{ij} < b_i + s_i$ . Then,  $(x, s')$  with  $s'_i = \sum_{j \in N} a_{ij}x_{ij} - b_i < s_i$  is feasible and has objective value  $\sum_{i \in M} \sum_{j \in N} c_{ij}x_{ij} + \sum_{i \in M} c_i \cdot s'_i < \sum_{i \in M} \sum_{j \in N} c_{ij}x_{ij} + \sum_{i \in M} c_i \cdot s_i$ . Therefore,  $(x, s)$  can not be optimal.  $\square$

Thus,  $s_i$  can be replaced in the objective by  $\sum_{j \in N} a_{ij}x_{ij} - b_i$  while (4) is left out. After regrouping the objective function we obtain:

$$-\sum_{i \in M} c_i b_i + \min \sum_{i \in M} \sum_{j \in N} (c_{ij} + c_i a_{ij}) x_{ij} \quad (6)$$

$$\text{s.t. } \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \quad (7)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N \quad (8)$$

Problem (6) to (8) is solvable in linear time by setting  $x_{ij} = 1$  for each tuple  $(i, j)$  with  $j \in N$  and  $i = \arg \min_{i' \in M} c_{i'j}$ , using an arbitrary tie breaker if needed.

#### 3.2 Capacity Adjustment with Upper Bounds

Assume,  $l_i \leq -b_i$  for each agent and there is an  $i', i' \in M$ ,  $u_{i'} \leq W_{i'} - b_{i'}$ . Here again, for each solution constraint (2) is tight, see lemma 1. Thus we can replace  $s_i$  by  $\sum_{j \in N} a_{ij}x_{ij} - b_i$  in (1) and (4)  $\forall i \in M$ , where the lower bounds  $l_i$  are already left out. Now the problem can be stated as follows:

$$-\sum_{i \in M} c_i b_i + \min \sum_{i \in M} \sum_{j \in N} (c_{ij} + c_i a_{ij}) x_{ij} \quad (9)$$

$$\text{s.t. } \sum_{i \in M} x_{ij} = 1 \quad \forall i \in M \quad (10)$$

$$\sum_{j \in N} a_{ij}x_{ij} \leq b_i + u_i \quad \forall i \in M \quad (11)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N \quad (12)$$

This is a standard GAP with modified cost and adjusted capacities. Obviously, for all  $i'' \in M$ ,  $u_{i''} > W_{i''} - b_{i''}$ , constraint (11) can be left out, too.

### 3.3 Capacity Adjustment with Lower Bounds

Now, we have an  $i', i' \in M$ ,  $l_{i'} \geq -b_{i'}$  and  $u_i \geq W_i - b_i$  for each  $i \in M$ . Here, (4) is simply replaced by  $l_i \leq s_i \forall i \in M$ , which does not change the underlying problem.

As no further reduction is possible here, we propose to transfer the concepts, developed for the knapsack problem with a continuous variable, KPC for short, see Büther and Briskorn [1]: We implement the binary coding of the adjustments for the eGAP.

First, we replace the capacity adjustment  $s_i$  by  $s_i = u_i - \bar{s}_i \forall i \in M$ , see Nauss [9]. The new variables  $\bar{s}_i$  represent the unused capacity adjustments and thus, they are bounded by  $0 \leq \bar{s}_i \leq u_i - l_i \forall i \in M$ . For illustration one can imagine, that in advance the capacities are extended at cost  $c_i \cdot u_i$  and then, during the optimization, the superfluous capacities can be sold which is expressed by  $\bar{s}_i > 0$ . Now, the continuous variables  $\bar{s}_i$  are substituted by binary code  $\bar{s}_i = \sum_{k_i \in D_i} e_{ik_i} y_{ik_i}$ , with  $D_i = \{1, \dots, d_i\} \forall i \in I$ . Here, binary variable  $y_{ik_i}$  equals one if and only if  $e_{ik_i}$  contributes to the sold capacity.

Now, we detail our requirements regarding  $d_i$  and  $e_{ik_i}$ , see Büther and Briskorn [1]:

- i Values of  $\bar{s}_i$  must not undershoot 0 or exceed  $u_i - l_i$ , respectively, that is  $0 \leq \sum_{k_i \in D_i} e_{ik_i} y_{ik_i} \leq u_i - l_i$  for each binary vector  $y_i = \{y_{i1}, \dots, y_{id_i}\}$ .
- ii Given a precision  $p \in \mathbb{N}$  for each value  $v_i \in [0, u_i - l_i]$  there must be a binary vector  $y_i$  such that  $|v_i - \sum_{k_i \in D_i} e_{ik_i} y_{ik_i}| \leq 10^{-p}$ .
- iii Regarding i and ii the number of additional binary variables  $d_i$  should be chosen as small as possible.

It is well-known, that any integer value can be stated by a binary representation. Thus, by choosing the values for  $e_{ik_i}$  appropriate, we can represent any integer number between 0 and  $u_i - l_i$ . If  $u_i - l_i = 2^r - 1$  with  $r \in \mathbb{N}$  then  $d_i = \log_2(u_i - l_i + 1) = r$  and  $e_{ik_i} = 2^{k_i-1}$  for each  $k_i \in D_i$  is the obvious choice. Otherwise, if  $u_i - l_i \neq 2^r - 1$  for all  $r \in \mathbb{N}$  we choose  $d_i = \lceil \log_2(u_i - l_i + 1) \rceil$ ,  $e_{ik_i} = 2^{k_i-1}$  for each  $k_i \in \{1, \dots, d_i-1\}$  and  $e_{id_i} = u_i - l_i - 2^{d_i-1} + 1$ .

Additionally, the concept can even be used approximately for real numbers for the adjustment  $s_i$ . Regarding the required precision  $p \in \mathbb{N}$  we apply the procedure described above to the value  $\lfloor 10^p \cdot (u_i - l_i) \rfloor \forall i \in M$ . After obtaining values  $e_{ik_i}$ , we divide them by  $10^p$ . Obviously, these transformed values suffice condition ii, see Müller-Bungart [7]. The number of additional variables is now  $d_i = \lceil \log_2(\lfloor 10^p(u_i - l_i) \rfloor + 1) \rceil$ . Therefore, the approximation and thus the quality of the upper bound can be improved arbitrarily by increasing  $p$ .

The reformulation, namely eGAP<sub>bin</sub>, can now be represented as follows:

$$\sum_{i \in M} c_i u_i + \min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} - \sum_{i \in M} \sum_{k_i \in D_i} c_i e_{ik_i} y_{ik_i} \quad (13)$$

$$\text{s.t. } \sum_{j \in N} a_{ij} x_{ij} + \sum_{k_i \in K_i} e_{ik_i} y_{ik_i} \leq b_i + u_i \quad \forall i \in M \quad (14)$$

$$\sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \quad (15)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N \quad (16)$$

$$y_{ik_i} \in \{0, 1\} \quad \forall i \in M, \forall k_i \in D_i \quad (17)$$

**Lemma 2.** *The model formulation  $eGAP_{bin}$  can be transferred to an equivalent standard GAP.*

The basic idea of our proof is to extend the  $eGAP_{bin}$  by adding artificial variables, which are then used to build the required constraints for a standard GAP.

*Proof.* First, the artificial binary variables  $y_{i'k_i}, \forall i' \in M, \forall k_i \in D_i$ , with  $i' \neq i$  are introduced. These variables correspond to the extension  $u_i$  at agent  $i$  and they enable us to "sell" this capacity extension at agent  $i'$ . Obviously, for these variables we require  $e_{i'k_i} = 0 \forall i' \neq i$  to have no influence neither on the constraints nor on the objective function value, as such a selling is not admissible. Then, we can formulate analog to (15):

$$\sum_{i' \in M} y_{i'k_i} = 1 \quad \forall i \in M, \forall k_i \in D_i \quad (18)$$

Apparently, combining (13) to (17) with the artificial variables and (18), we get a standard GAP.

For each solution of the GAP, one can find a corresponding solution for the  $eGAP_{bin}$ , having the same assignment of jobs and the same objective function value and the other way round.

First, for the GAP, if we have for an arbitrary  $y_{ik_i} = 0, i \in M$ , then (18) reduces to  $\sum_{i' \in M \setminus \{i\}} y_{i'k_i} = 1$ . As all artificial variables have coefficients of zero, an arbitrary of them will take on the value 1 and the remaining will be 0. Therefore, an equivalent solution for the  $eGAP_{bin}$  can be found by simply setting  $y_{ik_i} = 0$  and ignoring the artificial variables. Otherwise, if we have  $y_{ik_i} = 1, i \in M$ , for the GAP, then (18) reduces to  $\sum_{i' \in M \setminus \{i\}} y_{i'k_i} = 0$  and all artificial variables  $y_{i'k_i}, i' \in M \setminus \{i\}$ , are zero. In this case we have a corresponding solution for the  $eGAP_{bin}$ , by setting  $y_{ik_i} = 1$ , too.

Now, given a solution for the  $eGAP_{bin}$ , if  $y_{ik_i} = 0, i \in M$ , one gets a corresponding solution for the GAP by setting an arbitrary artificial variable  $y_{i'k_i} = 1, i' \in M \setminus \{i\}$ , and the remaining in (18) to 0. Therefore, one can find  $M - 1$  solutions for the GAP, having the same objective function value. Otherwise, if  $y_{ik_i} = 1, i \in M$ , all artificial variables  $y_{i'k_i}, i' \in M \setminus \{i\}$ , of the GAP have to be set to 0 and  $y_{ik_i} = 1$ , to generate the corresponding solution.  $\square$

Combining the two cases for the "real" variables for the adjustment in the GAP, either  $y_{ik_i}$  or any corresponding artificial variable is set to one or for short  $y_{ik_i} \in \{0, 1\}$ , as required in the  $eGAP_{bin}$ , too. Summarized, fulfilling (18) in the GAP is equivalent to ensuring (17) in the  $eGAP_{bin}$ .

Due to the binary coding, the number of variables increases. The original model,  $eGAP$ , has  $M \cdot N + M$  variables, where the  $eGAP_{bin}$  already has  $M \cdot N + \sum_{i \in M} d_i$  variables, with  $d_i = \lceil \log_2(u_i - l_i + 1) \rceil \geq 1$ . If we extend the range of the adjustment  $s_i$  to real numbers, the value of  $d_i$  increases to  $d_i = \lceil \log_2(10^p \cdot (u_i - l_i) + 1) \rceil$ .

Thus, the equivalent GAP for the  $eGAP_{bin}$  contains  $N'$  jobs, where  $N' = N + \sum_{i \in M} d_i$ , corresponding to the number of real jobs and the number of pseudo-jobs for the capacity adjustment.

Since we reduce the solution space to discrete values of  $\tilde{s}$  within  $[0, u_i - l_i]$ , where the domain of  $\tilde{s}_i$  is a subset of the domain of  $\bar{s}_i$ , the solution obtained via  $eGAP_{bin}$  is a feasible solution for the original model  $eGAP$ , too. Thus, the solution to  $eGAP_{bin}$  is an upper bound for the

optimal solution to eGAP. Given an optimal solution  $(x, \bar{s})$  to  $\text{eGAP}_{bin}$  the upper bound can be strengthened if  $b_i + l_i < \sum_{j \in N} a_{ij} x_{ij} < b_i + u_i - \bar{s}_i$ . This appears, if the necessary adjustment is not presentable by binary coding regarding the precision  $p$ . Then,  $\bar{s}_i$  can be increased to  $\lceil \bar{s}_i \rceil^= = b_i + u_i - \sum_{j \in N} a_{ij} x_{ij}$ , according to lemma 1. In the following we analyze the gap between the upper bound by  $\text{eGAP}_{bin}$  and the optimal solution to eGAP.

**Theorem 1.** *The gap  $\Delta$  between the upper bound obtained by  $\text{eGAP}_{bin}$  and the optimal objective value according to eGAP can not be larger than  $\sum_{i \in M} c_i \cdot 10^{-p}$ .*

*Proof.* Let  $\lfloor \bar{s}_i \rfloor_{\bar{s}}$  denote the binary code which is closest to but not larger than  $\bar{s}_i$ . Then, given the optimal solution  $(x, \bar{s})$  to eGAP and the optimal solution  $(x', \bar{s})$  to  $\text{eGAP}_{bin}$  the following holds.

$$\sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} - \sum_{i \in M} c_i \cdot \bar{s}_i \leq \sum_{i \in M} \sum_{j \in N} c_{ij} x'_{ij} - \sum_{i \in M} c_i \cdot \lceil \bar{s}_i \rceil^= \quad (19)$$

$$\leq \sum_{i \in M} \sum_{j \in N} c_{ij} x'_{ij} - \sum_{i \in M} c_i \cdot \bar{s}_i \quad (20)$$

$$\leq \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} - \sum_{i \in M} c_i \cdot \lfloor \bar{s}_i \rfloor_{\bar{s}} \quad (21)$$

where the constant  $\sum_{i \in M} c_i \cdot u_i$  has been left out for simplification.

Relation (19) expresses that the improved optimal solution to  $\text{eGAP}_{bin}$  is an upper bound to the optimal solution of eGAP. Obviously, this improved solution has no higher value than the optimal solution to  $\text{eGAP}_{bin}$  itself, giving (20). Representation of  $(x, \bar{s})$  in terms of  $\text{eGAP}_{bin}$  can not have a smaller value than  $(x', \bar{s})$ , as otherwise this solution would have been found with  $\text{eGAP}_{bin}$  as well, therefore (21) is valid.

Apparently, regarding requirement ii we get:

$$\Delta \leq \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{i \in M} c_i \cdot \bar{s}_i - \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} + \sum_{i \in M} c_i \cdot \lfloor \bar{s}_i \rfloor_{\bar{s}} \quad (22)$$

$$= \sum_{i \in M} c_i (\bar{s}_i - \lfloor \bar{s}_i \rfloor_{\bar{s}}) \leq \sum_{i \in M} c_i \cdot 10^{-p}. \quad (23)$$

□

As we can see from theorem 1 we can always obtain an upper bound (and a corresponding heuristic solution) which is arbitrarily close to the optimal objective value by increasing  $p$ . Obviously, if all coefficients of the jobs in the capacity constraints are integer numbers, the adjustment will only take on integer numbers, too, according to lemma 1. Thus, with  $p = 0$  our reformulation will always deliver an optimal solution. This leads to the open question whether we can choose  $p$  such that we always obtain an optimal solution by solving  $\text{eGAP}_{bin}$ .

Since the  $\text{eGAP}_{bin}$  can be transferred to an equivalent standard GAP, all known efficient algorithms for the GAP can be applied one-to-one. Every feasible solution will be feasible to the underlying eGAP, too. Additionally, one can now use standard cover cuts to fasten the corresponding lp relaxation, see Wolsey [15]. In figure 1 an extended capacity is given for an agent  $i$ ,  $i \in M$ . Apparently, all original jobs,  $x_{ij}$ , can now be assigned to this specific agent. But regarding the original jobs and the pseudo-jobs,  $y_{ik_i}$ , not all can be assigned to the agent at the same time and we get a typical knapsack constraint, which enables cover cuts.

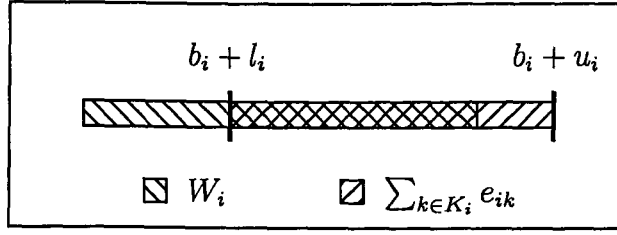


Figure 1: Illustration of  $W_i$  and a maximal capacity extension at agent  $i$

Note, as only a lower bound is restrictive, the capacities can be extended arbitrarily and every assignment of jobs to agents is feasible. Thus there exists  $m^n$  feasible assignments, which might result in high computation times, if heuristics based on local search are used for analyzing the solution space, see section 1.

### 3.4 Capacity Adjustment with Lower and Upper Bounds

In this case, our reduction of the model formulation analog to section 3.1 is not possible and we propose to use at once the binary coding of the adjustment as presented in section 3.3. Thus we obtain again an heuristic solution where the objective function value is an upper bound for the optimal one. As we have two restrictive bounds for the adjustment now, we have to focus on the appropriate choice of the  $e_{ik_i}$  to ensure that  $\sum_{k_i \in D_i} e_{ik_i} \leq u_i - l_i$ , with  $D_i = \{1, \dots, d_i\} \forall i \in I$ , see the previous section and Büther and Briskorn [1] for further details. The obtained model is a standard GAP, too, and the results of the previous section can be transferred analogously.

## 4 Capacity Adjustment with two Variables

This is a generalization of the eGAP, where we have two continuous variables  $s_i^-$  and  $s_i^+$  for each capacity constraint. The first variable,  $s_i^-$ ,  $i \in M$ ,  $0 \leq s_i^- \leq u_i^-$ , is used for reducing the capacity, the second,  $s_i^+$ ,  $i \in M$ ,  $0 \leq s_i^+ \leq u_i^+$ , for extending it, respectively. Assume, we can use not needed capacities for other projects, thus each unit of unused capacity reduces the total cost, denoted by  $c_i^-$ , while an extension still increases the cost, denoted by  $c_i^+$ . The variables  $s_i^+$  are identical to  $s_i$  in the previous sections. Now, the double elastic GAP, deGAP for short, can be expressed as follows:

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} - \sum_{i \in M} c_i^- s_i^- + \sum_{i \in M} c_i^+ s_i^+ \quad (24)$$

$$\text{s.t. } \sum_{j \in N} a_{ij} x_{ij} + s_i^- \leq b_i + s_i^+ \quad \forall i \in M \quad (25)$$

$$\sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \quad (26)$$

$$0 \leq s_i^- \leq u_i^- \quad \forall i \in M \quad (27)$$

$$0 \leq s_i^+ \leq u_i^+ \quad \forall i \in M \quad (28)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N \quad (29)$$

We do not claim equality in (25): If the upper bound  $u_i^-$  is restrictive for an arbitrary  $i' \in M$ , we might still use less capacity than  $b_i - u_i^-$  at this agent  $i'$  and "waste" unused capacity. A real-life application for such a model might appear, if we try to solve problems at capital markets, where we can borrow or place money, respectively, with different interest rates.

Obviously, we have to ensure, that  $c_i^- < c_i^+ \forall i \in M$ . Otherwise, we would be able to gain a profit by extending the capacity by  $s_{2i} > 0$  and "sell" this capacity at once,  $s_i^- > 0$ . One unit of this nonsensical adjustment would be stated by  $c_i^+ - c_i^- < 0$ . From this it follows that only one of both variables for the capacity adjustment will be larger than its lower bound. Therefore, an idea might be to split the problem into two subproblems for each agent, the first requiring  $s_i^- \geq 0 \wedge s_i^+ = 0$  and the second requiring  $s_i^- = 0 \wedge s_i^+ \geq 0$ . Unfortunately, this results into  $2^M$  problems, each of them being an eGAP, which is known to be NP-hard.

Note, if  $c_i^- = c_i^+ \forall i \in M$ , we get the pure eGAP with  $s_i$  bounded by  $-u_i^- \leq s_i \leq u_i^+ \forall i \in M$ , see section 3.4. Therefore, we showed in section 3.4 that the binary coding of the adjustment is an appropriate technique. If  $c_i^- \neq c_i^+ \forall i \in M$ , we propose to use the binary coding, too. First, we replace  $s_i^+$  by  $s_i^+ = u_i^+ - \bar{s}_i^+ \forall i \in M$ , see Nauss [9], where  $\bar{s}_i^+$  is limited by  $0 \leq \bar{s}_i^+ \leq u_i^+ \forall i \in M$ . Now, all continuous variables  $\bar{s}_i^+$  as well as  $s_i^-$  are substituted by binary coding:  $\bar{s}_i^+ = \sum_{k_i \in D_i^+} e_{ik_i}^+ y_{ik_i}^+$  with  $D_i^+ = \{1, \dots, d_i^+\}$  and  $\bar{s}_i^- = \sum_{k_i \in D_i^-} e_{ik_i}^- y_{ik_i}^-$  with  $D_i^- = \{1, \dots, d_i^-\}$ ,  $i \in M$ . Here again, the determination of the needed values has to ensure requirements i to iii, see section 3.3. Now, the problem can be stated as:

$$\sum_{i \in M} c_i^+ u_i^+ + \min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij} - \sum_{i \in M} \sum_{k_i \in D_i^-} c_i^- e_{ik_i}^- y_{ik_i}^- - \sum_{i \in M} \sum_{k_i \in D_i^+} c_i^+ e_{ik_i}^+ y_{ik_i}^+ \quad (30)$$

$$\text{s.t. } \sum_{j \in N} a_{ij} x_{ij} + \sum_{k_i \in D_i^-} e_{ik_i}^- y_{ik_i}^- + \sum_{k_i \in D_i^+} e_{ik_i}^+ y_{ik_i}^+ \leq b_i + u_i^+ \quad \forall i \in M \quad (31)$$

$$\sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \quad (32)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N \quad (33)$$

$$y_{ik_i}^- \in \{0, 1\} \quad \forall i \in M, \forall k_i \in D_i^- \quad (34)$$

$$y_{ik_i}^+ \in \{0, 1\} \quad \forall i \in M, \forall k_i \in D_i^+ \quad (35)$$

This again can be transferred to an equivalent standard GAP, which enables us to use standard approaches.

By adding several additional variables for extending or reducing the capacity with different cost and profits, we can model a cascaded cost function for the adjustment. Of course the values for the cost have to be chosen such that a generation of profit by buying and selling capacities at the same time is prevented. For such a problem formulation, of course, our binary coding can be used, too. A real life example can be found at capital markets, where the interest rate increases if the budget is extended more than a fixed value as the risk increases.

If we aim for a balanced capacity consumption for all agents, we get a modification of the deGAP. Now, each unit of unused capacity  $s_i^-$ , for each  $i \in M$ , causes costs. Thus, we strictly need equality in constraint (25), as otherwise  $s_i^-$  would always be at its lower bound 0,  $\forall i \in M$ . A related work can be found in Nauss [9]. Certainly, for this problem, the binary coding of the adjustments can be used, too. But, due to equality in constraint (25), we do not get a standard GAP. To our knowledge, the best solution method so far is an improved branch-and-bound algorithms, developed by Nauss [9]. Note, even if standard lagrangean relaxations are used on (26) for obtaining lower bounds on the objective, the resulting problem can not be decomposed into standard knapsack problems as it can be done for the GAP, see e.g. Yagiura et al. [17]. Due to equality in (25), the algorithms, developed by Volgenant and Marsman [13], Kaufman et al. [5] or Ram and Sarin [11] have to be used, which are more expensive regarding the computational efforts than e.g. the combo algorithm for the standard knapsack problem by Martello et al. [6].

## 5 Computational Results

After giving the theoretical background in the previous sections, we now present some computational results to show the efficiency of the developed reformulations with the binary coding of the adjustments. The algorithms are coded in  $C^{++}$  and executed on an Intel Pentium D processor with 1GB RAM and 2.80 GHz clockpulse using the operating system Windows XP. In order to compare the different formulations, continuous variables vs. binary representation, we used the commercial solver ILOG CPLEX 10.1 with standard settings.

There are five well-known types of benchmark instances for the GAP, called types A, B, C, D, and E, see Yagiura et al. [16]. These have been modified to eGAP according to Nauss [9] and solved once with continuous variables (eGAP) and once with the binary coding (eGAP<sub>bin</sub>). The adjustments are only limited by lower bounds  $l_i = 0$  for each  $i \in M$ . We set three different timelimits: 60s, 600s and 3600s, respectively, to compare the obtained bounds for the objectives of the different formulations. The results can be found in table 1 to 3. Note, that all coefficients for the capacity consumption are integer numbers for these instances, thus solving the binary reformulation can always deliver an optimal solution.

In table 1, the first column represents the instance type, where altogether 57 instances have been tested, split into 6–6–15–15–15 according to types A–B–C–D–E. For the next columns, eGAP and eGAP<sub>bin</sub> identify which formulation has been used. The values in the table give the percentage of the instances, that were proven to be solved optimally by CPLEX. Obviously, instances of type A are too easy and instances of type D too difficult to see any differences. But for all other types, the reformulation with binary coding of the adjustment is more often solved optimally.

Table 2 to 3 compare the best obtained lower and upper bound, respectively, using the same instances as before. Therefore, the first column represents again the instance type. Then,

Table 1: Optimality proven by CPLEX

	60s		600s		3600s	
	eGAP	eGAP <sub>bin</sub>	eGAP	eGAP <sub>bin</sub>	eGAP	eGAP <sub>bin</sub>
A	100%	100%	100%	100%	100%	100%
B	33%	83%	50%	83%	67%	100%
C	7%	27%	13%	40%	20%	53%
D	0%	0%	0%	0%	0%	0%
E	7%	13%	20%	40%	27%	47%

for all timelimits, there occur three columns: The first gives the percentage of identical bounds of both formulations. The second the percentage of instances, where the binary coding delivers a better bound and the third, where the original formulation provides the better bound. Regarding the results, we see again, that instances of type A are too easy. Note, for type A, although CPLEX has always proven optimality, the lower bound by eGAP<sub>bin</sub> are sometimes higher. This is founded on the stopping criteria of the branch and bound algorithm implemented in CPLEX, the corresponding upper bounds are always identical. For the other instance types, the binary coding clearly outperforms the equivalent formulation with continuous variables in generating lower bounds. The same yields mostly for the upper bounds, too, only for the instances of type D, both formulations seem to be equivalent.

Table 2: Best obtained lower bound

	60s			600s			3600s		
	ident	bin	con	ident	bin	con	ident	bin	con
A	83%	17%	0%	83%	17%	0%	83%	17%	0%
B	33%	67%	0%	50%	50%	0%	50%	50%	0%
C	13%	87%	0%	13%	80%	7%	20%	73%	7%
D	20%	80%	0%	20%	80%	0%	20%	80%	0%
E	0%	100%	0%	7%	93%	0%	7%	93%	0%

Table 3: Best obtained upper bound

	60s			600s			3600s		
	ident	bin	con	ident	bin	con	ident	bin	con
A	100%	0%	0%	100%	0%	0%	100%	0%	0%
B	83%	17%	0%	83%	17%	0%	83%	17%	0%
C	13%	80%	7%	27%	73%	0%	40%	60%	0%
D	0%	47%	53%	0%	67%	33%	7%	53%	40%
E	7%	60%	33%	13%	67%	20%	13%	73%	13%

Note that using the continuous variables, CPLEX ran in 51% out of memory, when a timelimit of 3600s was used. For the binary representation, the same occurs in only 23% of the instances.

Knowing well, that the chosen instances are just a small sample, we notice:

- For the given instances, the commercial server CPLEX can handle better the binary coding for producing lower bounds, probably due to the well implemented cut generation to strengthen the lp-relaxation.

- Regarding the absolute values for the obtained lower bounds, they only increase a little bit for the binary coding, if the timelimit increases. But for most difficult instances, the obtained lower bounds of the  $eGAP_{bin}$  using 60s are still better or equal than that obtained by eGAP using 3600s.
- The used instances are in part quiet easy to solve given a timelimit of 3600s. Thus in future we will have to test different cost for the adjustment of the capacity and in general adjustments limited by lower as well as upper bounds.

Summarizing, although the binary representation offers more variables in general, see section 3.3, the corresponding instances can be handled better by CPLEX regarding the computation times and obtained lower as well as upper bounds.

## 6 Conclusions and Outlook

In our previous work, see Büther and Briskorn [1], we proposed some useful reductions of the KPC to a standard knapsack problem, amongst others by the binary representation of the continuous variable. In this work, we transferred this concept from the KPC to the eGAP. We proofed, that the resulting problem can be transferred to an equivalent standard GAP. This is quite easy at the cost of getting only an heuristic solution, which has an objective function value theoretically arbitrary near to the optimal one. In the future, we will combine the eGAP, using the binary representation, with state-of-the-art algorithms for the GAP to handle hard instances, where CPLEX fails due to the lack of memory, see section 5. Additionally, we will generate more eGAP instances to test, whether the trend holds on, that the integer problem outperforms the mixed integer formulation using standard solvers.

Obviously, the replacement of a continuous variable by several binary ones can be transferred to several other mixed integer problems, too, e.g. the elastic single source capacitated facility location problem, eSSCFLP for short. This problem consists in selecting plant sites from a finite set of potential sites and in allocating customer demands in such a way that the operating and transportation cost are minimized while regarding the adjusted capacities. Therefore, if the optimal facilities are selected, finding the optimal assignment of customers results into an eGAP. One economic interpretation for the eSSCFLP is the optimal selection of employees. While regarding the (extended) working time for each employee, we have to ensure that all jobs are fulfilled: overtime cost vs cost for hiring new employees.

Unfortunately, already for the eGAP, it was not possible to determine an optimal solution straightforwardly, like it can be done for the KPC in some special cases, for further details we refer to Büther and Briskorn [1]. For more complex problems, regarding the structure and number of the constraints as well as the structure of the objective function, probably the same yields, as long as the problem can be reduced to the eGAP. Thus, like presented for the eGAP, only the binary replacement of the continuous variables can be transferred to these mixed integer problems.

## References

- [1] M. Büther and D. Briskorn. Reducing the 0-1 knapsack problem with a single continuous variable to the standard 0-1 knapsack problem. Technical report, Christian-Albrechts-Universität at Kiel, 2007.
- [2] S. Ceria, C. Cordier, H. Marchand, and L. A. Wolsey. Cutting planes for integer programs with general integer variables. *Mathematical Programming*, 81:201–214, 1998.
- [3] A. Fügenschuh and A. Martin, 2005. Computational integer programming and cutting planes, in: K. Aardal, G.L. Nemhauser and R. Weismantel (eds.), *Handbooks in Operations Research and Management Science*, Vol. 2, 69–121.
- [4] F. Glover, M. Laguna, and R. Mart. Fundamentals of scatter search and path relinking. 2000.
- [5] L. Kaufman, F. Plastria, and S. Tubeecx. Zero-one knapsack problem with equality constraint. *European Journal of Operational Research*, 19(3):384–389, 1985.
- [6] S. Martello, D. Pisinger, and P. Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999.
- [7] M. Müller-Bungart. Revenue management with flexible products. *Lecture Notes in Economics and Mathematical Systems*. Springer Berlin, 2007.
- [8] R. M. Nauss. Solving the generalized assignment problem: An optimizing and heuristic approach. *INFORMS Journal on Computing*, 15(3):249–266, 2003.
- [9] R. M. Nauss. The elastic generalized assignment problem. *Journal of the Operational Research Society*, 55:1333–1341, 2004.
- [10] G. L. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1999.
- [11] B. Ram and S. Sarin. A algorithm for the 0-1 equality knapsack problem. *Journal of the Operational Research Society*, 39(11):1045–1049, 1988.
- [12] A. Schrijver. *Combinatorial Optimization – Polyhedra and Efficiency*. Springer, 2003.
- [13] A. Volgenant and S. Marsman. A core approach to the 0-1 equality knapsack problem. *Journal of the Operational Research Society*, 49(1):86–92, 1998.
- [14] L. A. Wolsey. Strong formulations for mixed integer programs: valid inequalities and extended formulations. *Mathematical Programming*, 97(1-2):423–447, 2003.
- [15] L. A. Wolsey. *Integer Programming*. Wiley, New York, 1998.
- [16] M. Yagiura, T. Ibaraki, and F. Glover. An ejection chain approach for the generalized assignment problem. *INFORMS Journal on Computing*, 16(2):133–151, 2004.
- [17] M. Yagiura, T. Ibaraki, and F. Glover. A path relinking approach for the generalized assignment problem. *European Journal of Operational Research*, 169:548–569, 2006.