Math Meth Oper Res (2011) 73:19–27
DOI 10.1007/s00186-010-0331-4

On generalized balanced optimization problems

**Lara Turner · Abraham P. Punnen ·
Yash P. Aneja · Horst W. Hamacher**

Received: 31 March 2010 / Accepted: 17 September 2010 / Published online: 16 October 2010

Abstract In the generalized balanced optimization problem (GBaOP) the objective value $\max_{e \in S} |c(e) - k \max(S)|$ is minimized over all feasible subsets S of $E = \{1, \ldots, m\}$. We show that the algorithm proposed in Punnen and Aneja (Oper Res Lett 32:27–30, 2004) can be modified to ensure that the resulting solution is indeed optimal. This modification is attained at the expense of increased worst-case complexity, but still maintains polynomial solvability of various special cases that are of general interest. In particular, we show that GBaOP can be solved in polynomial time if an associated bottleneck problem can be solved in polynomial time. For the solution of this bottleneck problem, we propose two alternative approaches.

Keywords Combinatorial optimization · Balanced optimization ·
Bottleneck problems

L. Turner (✉) · H. W. Hamacher
Department of Mathematics, Technical University of Kaiserslautern,
P. O. Box 3049, 67653 Kaiserslautern, Germany
e-mail: turner@mathematik.uni-kl.de

H. W. Hamacher
e-mail: hamacher@mathematik.uni-kl.de

A. P. Punnen
Department of Mathematics, Simon Fraser University Surrey, Central City,
250-13450 102nd AV, Surrey, BC V3T 0A3, Canada
e-mail: apunnen@sfu.ca

Y. P. Aneja
Odette School of Business, University of Windsor, Windsor, ON N9B 3P4, Canada
e-mail: aneja@uwindsor.ca

1 Introduction

Let $E = \{1, 2, \ldots, m\}$ be a finite set and let $c(e) \in \mathbb{R}$ be given costs of the elements $e \in E$. Let $F \subseteq 2^E$ be the set of feasible solutions such that $S \in F$ implies $|S| = n$ for some positive integer n. Then the *generalized balanced optimization problem* (GBaOP) is

$$\text{GBaOP} \qquad \underset{S \in F}{\text{Minimize}} \ \max_{e \in S} |c(e) - k\max(S)|$$

where $k\max(S)$ is the kth largest cost coefficient of solution S. Without loss of generality, we assume that all $c(e),\ e \in E$, are distinct. Otherwise, we could perturb the $c(e)$'s by very small amounts to make them distinct such that an optimal solution to the modified problem is also optimal for the original problem. GBaOP was introduced by Punnen and Aneja (2004) as an extension of the *balanced optimization problem* (BaOP) (Martello et al. 1984) and the *lexicographic balanced optimization problem* (LBaOP). While the algorithm for LBaOP presented in Punnen and Aneja (2004) is correct, the algorithm presented for GBaOP does not guarantee an optimal solution and can only be viewed as a heuristic. This is shown in the following example.

Example 1 Let $G = (V, E)$ be an undirected graph with costs $c(e)$ for all edges $e \in E$ as given in Fig. 1 and let F be the set of all spanning trees in G. In graph G all spanning trees S have cardinality $|S| = |V| - 1 = 3$. Choosing $k = 3$, GBaOP reduces to the classical balanced optimization problem for which tree $S^* = \{[1, 2], [2, 4], [3, 4]\}$ is optimal. The approach of Punnen and Aneja (2004) is to solve GBaOP sequentially for all $\alpha \in \{c(e),\ e \in E\}$ by restricting the set of feasible solutions to those trees in graph G with $\max_{e \in S} c(e) = \alpha$. Solving these problems as bottleneck problems over the original, i.e. unrestricted, feasible set with costs $|c(e) - \alpha|$ for all $e \in E$ is, however, not correct as our example shows. Setting $\alpha = 4$ and $\alpha = 7$ for which the restricted set of spanning trees is non-empty, the corresponding bottleneck problems compute tree $S = \{[1, 2], [1, 3], [3, 4]\}$ with $\max_{e \in S} c(e) = 7$. As such it is only feasible for the restricted GBaOP with $\alpha = 7$.

In this paper we show that the algorithm proposed in Punnen and Aneja (2004) can be modified to ensure that the resulting solution is indeed optimal. This modification is attained at the expense of increased worst-case complexity, but still maintains polynomial solvability of various special cases that are of general interest. In particular, we show that GBaOP can be solved in polynomial time if an associated bottleneck problem can be solved in polynomial time. For the solution of this bottleneck problem, we propose two alternative approaches.

2 Algorithm for GBaOP

For any real number α, we define

$$w_e(\alpha) = |c(e) - \alpha| \tag{1}$$

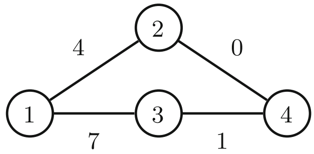

Fig. 1 Undirected graph G with costs $c(e)$

and consider the bottleneck problem

$$\text{BP}(\alpha) \qquad \underset{S \in F(\alpha)}{\text{Minimize}} \ \max\{w_e(\alpha) : \ e \in S\}$$

where $F(\alpha) = \{S \in F : \ k\max(S) = \alpha\}$. An optimal solution to BP(α) is denoted by $U(\alpha)$ and the problem GBaOP where F is replaced by $F(\alpha)$ is denoted by GBaOP(α).

Theorem 1 *If $F(\alpha) \neq \emptyset$ then $U(\alpha)$ is an optimal solution to GBaOP(α).*

Proof If $F(\alpha) \neq \emptyset$ then GBaOP(α) is feasible. Note that $k\max(U(\alpha)) = \alpha$ by definition of $F(\alpha)$. Choose any $S \in F(\alpha)$. Then

$$\begin{aligned} &\max_{e \in S} |c(e) - k\max(S)| = \max_{e \in S} |c(e) - \alpha| \\ &\quad \geq \max_{e \in U(\alpha)} |c(e) - \alpha| = \max_{e \in U(\alpha)} |c(e) - k\max(U(\alpha))|. \end{aligned}$$ □

Based on Theorem 1, we have the following scheme to solve GBaOP: For each $e \in E$ choose $\alpha = c(e)$ and solve BP($c(e)$) which produces an optimal solution $U(c(e))$ whenever $F(c(e))$ is non-empty. Choose e^* such that

$$\begin{aligned} &\min_{e \in E} \{ \max_{h \in U(c(e))} |c(h) - k\max(U(c(e)))| : \ F(c(e)) \neq \emptyset\} \\ &\quad = \max_{h \in U(c(e^*))} |c(h) - k\max(U(c(e^*)))|. \end{aligned} \tag{2}$$

Theorem 2 *$U(c(e^*))$ is an optimal solution to GBaOP. Further, GBaOP can be solved in polynomial time whenever BP(α) can be solved in polynomial time.*

Proof Note that $F = \bigcup_{e \in E} F(c(e))$. The result follows from Eq. (2) and Theorem 1. □

Theorems 1 and 2 are similar to the corresponding results in Punnen and Aneja (2004) except that we changed the definition of BP(α). In the next two subsections two approaches to solve $BP(\alpha)$ are presented. In the first one we assume distinct costs $c(e)$ for all $e \in E$ and a fixed cardinality for all feasible solutions $S \in F$, while we relax these assumptions in the second approach.

2.1 Solving BP(α): Version 1

To solve BP(α) we first consider the *color constrained bottleneck problem* (CCBP) which is a bottleneck version of the *color constrained combinatorial optimization* *problem* introduced by Hamacher and Rendl (1991). Let $w_1, w_2, \ldots, w_m$ be any given weights associated with the elements of E. Without loss of generality we assume that the w_e's are distinct for otherwise we could use ε-perturbations to construct an equivalent problem satisfying this property. Further, we assume that E is partitioned as $E = E_1 \cup E_2$ and define $w_{\max}(S) = \max\{w_e : e \in S\}$ for all $S \in F$. Then the color constrained bottleneck problem is to find a solution $S^* \in F$ such that $w_{\max}(S^*)$ is minimized and S^* contains exactly q elements from E_1 for a given integer q, $1 \le q \le n$. Setting

$$d_e = \begin{cases} 1 & \text{if } e \in E_1 \\ 0 & \text{otherwise} \end{cases}$$

CCBP can be formally stated as

$$\text{CCBP} \quad \text{Minimize } w_{\max}(S)$$
$$\text{Subject to}$$
$$S \in F, \quad \sum_{e \in S} d_e = q.$$

Thus CCBP is a variation of the constrained bottleneck problem studied by Berman et al. (1990) and Lee (1992) except that the additional constraint has 0–1 coefficients and needs to be satisfied as equality instead of inequality as required in Berman et al. (1990) and Lee (1992). This equality restriction brings in additional complexity to the problem.

For $Q(\beta) = \{e \in E : w_e \le \beta\}$ and $R(\beta) = \{S \in F : S \subseteq Q(\beta)\}$ we consider the feasibility problem FEAS(β): "Given a value β, does there exist an $S \in R(\beta)$ such that $\sum_{e \in S} d_e = q$?"

Since CCBP is essentially a bottleneck problem, using the binary search version of the threshold algorithm (Edmonds and Fulkerson 1970; Berman et al. 1990; Lee 1992), it can be solved as a sequence of $O(\log m)$ feasibility problems FEAS(β) for various choices of β. We assume that a feasibility oracle for FEAS(β) is available which computes a 'yes' or 'no' answer for a given input β. Further we assume that another oracle SOL(β) is available which computes a solution $S \in R(\beta)$ satisfying $\sum_{e \in S} d_e = q$ for a 'yes' instance of FEAS(β). A formal description of the procedure is given in Algorithm 1.

To illustrate our algorithm for CCBP, let G be an undirected graph and let F be the family of all spanning trees of G. Let $G(\beta)$ be the spanning subgraph of G containing all the edges of weight $w_e \le \beta$. Then the feasibility oracle FEAS(β) simply tests if $G(\beta)$ contains a spanning tree with exactly q elements from E_1 and if the answer is 'yes' SOL(β) produces such a spanning tree S in $G(\beta)$. In this case, FEAS(β) and SOL(β) reduce to a matroid intersection problem (intersection of a graphic matroid and a partition matroid) and hence can be solved in polynomial time. In general, if F is the family of the base system of a matroid, CCBP can be solved in polynomial time. Due to the interrelation of GBaOP and multicriteria optimization problems (see Turner (2011)) the matroid intersection algorithm of Camerini and Hamacher (1989) may be used for this purpose.

Algorithm 1 CCBP Algorithm

Input: A finite set $E = \{1, \dots, m\}$, the set of feasible solutions $F \subseteq 2^E$ (in compact form), the costs w_e for all $e \in E$ and the oracles FEAS(β) and SOL(β).
Output: An optimal solution to CCBP.
Renumber the elements of E (if necessary) so that $w_1 < w_2 < \dots < w_m$.
$\ell = 1$; $u = m$; $k = \ell$;
while $u - \ell > 0$ **do**
  $k = \lfloor \frac{(l+u)}{2} \rfloor$;
  if FEAS(w_k) outputs 'yes' **then** $u = k$; **else** $\ell = k + 1$;
end while
Output SOL(w_k).

Let us consider another example where G is a bipartite graph and F is the family of all perfect matchings of G. Let $G(\beta)$ be the spanning subgraph of G containing all the edges of weight $w_e \leq \beta$. Then the feasibility oracle FEAS (β) simply tests if $G(\beta)$ contains a perfect matching with exactly q elements from E_1 and if the answer is 'yes' SOL (β) produces such a matching S in $G(\beta)$. In this case, FEAS (β) and SOL (β) reduce to the *colored bipartite matching problem* (CBMP) (Karzanov 1987; Yi et al. 2002). This is known to be solvable in polynomial time on complete biparite graphs and other special classes of graphs (Leclerc 1988–1989), or if q is fixed. Polynomial time reductions between CBMP and other combinatorial problems with open complexity status can be found in Deĭneko and Woeginger (2006) and Błażewicz et al. (2007).

Finally we observe that our bottleneck problem BP($c(e)$) is precisely a CCBP. To enforce the constraint $k \max(S) = c(e)$ we have to make sure that a solution S of BP($c(e)$) satisfies the following conditions:

1. S contains $n - k$ elements of cost less than $c(e)$, and
2. S contains element e.

Consider the partition $E_1 \cup E_2 \cup \{e\}$ of E where $E_1 = \{h \in E : c(h) < c(e)\}$. To enforce that element e is in the solution we contract it to obtain the modified set $E' = E - \{e\}$ and the new family of feasible solutions $F' = \{S - \{e\} : S \in F, e \in S\}$. To make BP($c(e)$) a special case of CCBP, we choose $q = n - k$ and set $w_h = w_h(c(e))$ for all $h \in E$ as defined in (1). Then $S^* \in F$ is an optimal solution to CCBP on ground set E' with partition $E_1 \cup (E' - E_1)$ and family of feasible solutions F' if and only if $S^* \cup \{e\}$ is an optimal solution to BP($c(e)$).

In the spanning tree and matching examples considered before, this contraction operation corresponds to the standard edge contraction in graphs.

If we relax the assumption that all solutions $S \in F$ have cardinality n, we use a slight modification. For simplicity, however, we assume that the cardinality of any feasible solution $S \in F$ is at least k, i.e. $|S| \geq k$. To guarantee that $k \max(S) = c(e)$ is still satisfied, we require for any feasible solution S of BP($c(e)$) that

1. S contains $k - 1$ elements of cost greater than $c(e)$, and
2. S contains element e,

and solve a color constrained bottleneck problem with ground set $E' = E_1 \cup (E' - E_1)$, feasible solutions F' and $q = k - 1$ where $E_1 = \{h \in E : c(h) > c(e)\}$.

2.2 Solving BP(α): Version 2

In this section, we present an alternative way to solve BP(α). It is no longer necessary to assume a fixed cardinality for the feasible solutions $S \in F$ or pairwise different costs $c(e)$. To start with, we sort the elements in the ground set by non-decreasing costs, i.e. $E = \{e_1, \ldots, e_m\}$ such that

$$c(e_1) \leq c(e_2) \leq \cdots \leq c(e_m).$$

Using binary weights

$$d_j(e_i) = \begin{cases} 1 & \text{if } i > j \\ 0 & \text{otherwise} \end{cases}$$

as in Gorski and Ruzika (2009), we show how to fix an element $e_{j_k} \in E$ as kth largest cost element of solutions $S \in F$.

Theorem 3 *For an element $e_{j_k} \in E,\ j_k \in \{1, \ldots, m\}$, it holds that e_{j_k} is the kth largest cost element of solution S if and only if $\sum_{e_i \in S} d_{j_k}(e_i) = k-1$ and $\sum_{e_i \in S} d_{j_k-1}(e_i) = k$.*

Proof By definition of $d_j(e_i)$, the sums $\sum_{e_i \in S} d_{j_k}(e_i)$ and $\sum_{e_i \in S} d_{j_k-1}(e_i)$ count the number of elements in solution S with index strictly greater than j_k or $j_k - 1$, respectively. Since the elements in E are sorted by non-decreasing costs and any element $e \in E$ occurs at most once in a feasible solution $S \in F$, we can argue as follows:

Suppose that $\sum_{e_i \in S} d_{j_k}(e_i) \neq k - 1$. If $\sum_{e_i \in S} d_{j_k}(e_i) < k - 1$, there are at most $k-2$ elements with index greater than j_k. As element $e_{j_k} \in E$ is at most once in solution S, it can only be the $(k - 1)$st largest cost element or it is not contained in S. If $\sum_{e_i \in S} d_{j_k}(e_i) > k - 1$, the kth largest cost element has index strictly greater than j_k, so e_{j_k} cannot be the kth largest cost element of solution S. For $\sum_{e_i \in S} d_{j_k-1}(e_i) \neq k$, a similar contradiction can be shown.

Conversely, if $\sum_{e_i \in S} d_{j_k}(e_i) = k - 1$ and $\sum_{e_i \in S} d_{j_k-1}(e_i) = k$, there are exactly $k - 1$ elements with index greater than j_k and exactly k elements with index greater than $j_k - 1$. Obviously, the one additional element with index greater than $j_k - 1$ has index j_k and is the kth largest cost element in solution S. □

Using Theorem 3, BP(α) for $\alpha = c(e_{j_k})$ and $e_{j_k} \in E$ can be formulated as

$$\begin{array}{ll} \text{BP}(c(e_{j_k})) & \text{Minimize } \max\{w_{e_i}(c(e_{j_k})) : e_i \in S\} \\ & \text{Subject to} \\ & S \in F, \quad \sum_{e_i \in S} d_{j_k}(e_i) = k - 1, \quad \sum_{e_i \in S} d_{j_k-1}(e_i) = k \end{array} \tag{3}$$

where $w_{e_i}(c(e_{j_k}))$ is defined as in (1). Compared to CCBP in Sect. 2.1, this formulation has two constraints of type $\sum_{e \in S} d_e = q$ which may increase the complexity of the problem. The feasibility test FEAS(β) in the generic threshold algorithm for bottleneck problems applied to problem (3) is now: "Given a value β, does there exist an $S \in F$ with $w_{e_i}(c(e_{j_k})) \leq \beta$ for all $e_i \in S$ such that $\sum_{e_i \in S} d_{j_k}(e_i) = k - 1$ and $\sum_{e_i \in S} d_{j_k - 1}(e_i) = k$?" As in Algorithm 1, SOL(β) is the procedure returning a solution S with this property.

As special case of a polynomially solvable problem we consider a directed acyclic graph G with source s and sink t and F as the set of all directed paths from s to t. Then, BP($c(e_{j_k})$) can be interpreted as a bottleneck version of a *resource constrained shortest path problem* (RCSP) (Lawler 2001) with equality constraints. In this case, the feasibility test FEAS(β) and the solution oracle SOL(β) can be solved simultaneously. In fact, they are classical RCSPs with sum objective, but with equality constraints instead of inequality constraints. They can be shown to be solvable by a dynamic programming algorithm that runs in polynomial time. Details of the resulting algorithm as well as related problems can be found in Turner (2011).

A similar reasoning applies to the minimization 0–1 knapsack problem where E is a set of given items with costs $c(e)$ and profits $p(e)$ for all $e \in E$. BP($c(e_{j_k})$) is a bottleneck knapsack problem with multiple constraints including the original knapsack constraint $\sum_{e_i \in S} p(e_i) \geq P$ for a given profit P and the equality constraints according to (3) in order to ensure that e_{j_k} is the kth largest item in solution S. It can again be solved by a dynamic programming approach which is, however, in contrast to the preceding path problem, solvable only in pseudopolynomial time (Kellerer et al. 2004; Turner 2011).

3 Conclusion

We showed that GBaOP can be solved as a polynomial sequence of special bottleneck problems, correcting an error in Punnen and Aneja (2004).

The solution approach can be extended to the more general problem

$$\underset{S \in F}{\text{Minimize}} \; \max_{e \in S} f(c(e) - k \max(S))$$

where $f : \mathbb{R} \to \mathbb{R}$ is a given real-valued function. The only difference is that we need to modify the definition of the weights in (1) to $w_e(\alpha) = f(c(e) - \alpha)$. An interesting special case of this generalization is when $f(c(e) - k \max(S)) = (c(e) - k \max(S))^{2p}$ for a positive integer p. In this case, the problem is equivalent to GBaOP in the sense that they have the same set of optimal solutions. Similar generalizations can be found in monotonic and stochastic balanced optimization. In Ţigan et al. (2005, 2008), the following problems

$$\underset{S \in F}{\text{Minimize}} \; f(\min_{e \in S} c(e), \max_{e \in S} d(e))$$

and

$$\underset{S \in F}{\text{Maximize}} \; g(\min_{e \in S} c(e), S)$$

are considered for given costs $c(e)$ and $d(e)$, where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is assumed to be strictly increasing and decreasing in the first and second argument, respectively, and where $g(\cdot, S) : \mathbb{R} \to \mathbb{R}$ is assumed to be strictly increasing.

Another problem which is closely related to GBaOP is the *k-balanced optimization problem* (k-BaOP) which is defined as

$$k\text{-BaOP1} \qquad \underset{S\in F}{\text{Minimize}}\ (\max_{e\in S} c(e) - k \max(S))$$

or

$$k\text{-BaOP2} \qquad \underset{S\in F}{\text{Minimize}}\ (k \max(S) - \min_{e\in S} c(e)).$$

It can be considered as a special case of the more general *universal combinatorial optimization problem* (Univ-COP) introduced in Turner (2011) and can be solved by a combination of threshold algorithms for balanced optimization problems (Martello et al. 1984) and k-max optimization problems (Gorski and Ruzika 2009).

Acknowledgments We thank two anonymous referees for pointing out minor errors and suggesting improvements in a first version of the paper. This research was in part supported by the Deutsche Forschungsgemeinschaft, Graduiertenkolleg “Mathematik und Praxis” (Lara Turner), NSERC discovery grants (Abraham P. Punnen), and Ministerium fuer Forschung und Technologie funding the project REPKA (Horst W. Hamacher).

References

Berman O, Einav D, Handler G (1990) The constrained bottleneck problem in networks. Oper Res 38: 178–181

Błażewicz J, Formanowicz P, Kasprzak M, Schuurman P, Woeginger GJ (2007) A polynomial time equivalence between DNA sequencing and the exact perfect matching problem. Discrete Optim 4:154–162

Camerini PM, Hamacher HW (1989) Intersection of two matroids: (condensed) border graphs and ranking. SIAM J Discrete Math 2:16–27

Deĭneko VG, Woeginger GJ (2006) On the robust assignment problem under a fixed number of cost scenarios. Oper Res Lett 34:175–179

Edmonds J, Fulkerson DR (1970) Bottleneck extrema. J Combin Theory 8:299–306

Gorski J, Ruzika S (2009) On k-max-optimization. Oper Res Lett 37:23–26

Hamacher HW, Rendl F (1991) Color constrained combinatorial optimization problems. Oper Res Lett 10:211–219

Karzanov AV (1987) Maximum matching of given weight in complete and complete bipartite graphs. Cybern Syst Anal 23:8–13

Kellerer H, Pferschy U, Pisinger D (2004) Knapsack problems. Springer, Berlin

Lawler EL (2001) Combinatorial optimization: networks and matroids. Dover Publications, Mineola

Leclerc M (1988–1989) Optimizing over a slice of the bipartite matching polytope. Discrete Math 73: 159–162

Lee J (1992) On constrained bottleneck extrema. Oper Res 40:813–814

Martello S, Pulleyblank WR, Toth P, de Werra D (1984) Balanced optimization problems. Oper Res Lett 3:275–278

Punnen AP, Aneja YP (2004) Lexicographic balanced optimization problems. Oper Res Lett 32:27–30

Ţigan Ş, Iacob EM, Stancu-Minasian IM (2005) Monotonic balanced optimization problem. Ann Tiberiu Popoviciu Seminar Funct Equ Approx Convex 3:183–197

Ţigan Ş, Stancu-Minasian IM, Coman I, Iacob ME (2008) On some stochastic balanced optimization problems. Ann Tiberiu Popoviciu Seminar Funct Equ Approx Convex 6:105–127

Turner L (2011) Universal combinatorial optimization problems. PhD thesis, Technical University of Kaiserslautern (to appear)

Yi T, Murty KG, Spera C (2002) Matchings in colored bipartite networks. Discrete Appl Math 121:261–277