Global Optimization and Multi Knapsack : a Percolation Algorithm

Dominique Fortin, Ider Tsevendorj

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Global Optimization and Multi Knapsack : a percolation algorithm

Dominique Fortin, Ider Tsevendorj

N° 3912

THÈME 4

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--3912--FR

INRIA ROCQUENCOURT

Global Optimization and Multi Knapsack : a percolation algorithm

Dominique Fortin, Ider Tsevendorj

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Adopt

Rapport de recherche n° 3912 — — 19 pages

Abstract: Since the standard multi knapsack problem max $\langle c, x\rangle$ s.t. $Ax \leq b, x \in \{0,1\}^n$, may be rewritten as a reverse convex problem, we present a global optimization approach. It is known from solving high dimensional nonconvex problems that pure cutting plane methods may fail and Branch&Bound is impractical, due to a large duality gap. On the other hand, a sufficient optimality condition-based strategy does not help much because it requires generating all level set points, an intractable problem. Therefore, we propose to combine both a cutting plane method and a sufficient optimality condition together with a random generation of level set points where the number of points is limited by a tabu list to prevent re-examination of the same level set area. Experiments show that we end up with a small duality gap permitting a subsequent Branch&Bound for reasonable sized instances.

Key-words: global optimization, multi knapsack, cutting planes, level set, geodesic

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Optimisation globale et Multi Knapsack: un algorithme de percolation

Résumé : Puisque le multi knapsack max $\langle c, x\rangle$ s.t. $Ax \leq b, x \in \{0,1\}^n$ se réécrit classiquement sous forme de problème convexe sous contrainte concave, nous détaillons une approche *optimisation globale*. Il est connu que pour des problèmes non convexes de grande taille, les méthodes de plans sécants peuvent échouer, de même que le *Branch&Bound* devient impratiquable, en raison d'un saut de dualité important. D'un autre côté, une stratégie basée sur une condition suffisante d'optimalité n'aide pas beaucoup puiqu'elle requiert la génération de tous les points des courbes de niveau. En conséquence, nous proposons une combinaison de plans sécants et d'une condition suffisante d'optimalité pour une génération aléatoire de points des courbes de niveau pour laquelle le nombre de points reste limité grâce à une liste tabou permettant d'éviter l'examen de points de la même zone de courbe de niveau, plusieurs fois. Les résultats expérimentaux montrent que nous terminons avec un faible saut de dualité permettant une résolution exacte par *Branch&Bound* pour des instances de taille raisonnable.

Mots-clé : optimisation globale, multi knapsack, plans sécants, courbe de niveau, géodésique

1. INTRODUCTION

Given an $m \times n$ matrix A with real nonnegative entries, a nonnegative weight vector $c = (c_1, \ldots, c_n)$ and a nonnegative supply vector $b = (b_1, \ldots, b_m)$, the classical multi knapsack problem is given by:

$$\begin{aligned} & \max\ \langle c, x\rangle \\ \text{s.t.}\ & Ax \leq b \\ & x \in \{0,1\}^n \end{aligned} \tag{MKP}$$

where each row $a^i \in R^n$ of A with corresponding b_i is a so-called knapsack constraint and we use angles for standard innerproduct; it is customary to assume each unit vector e_i of canonical basis of $\mathbb{R}^n$ feasible for (MKP) in order to deal with full dimensional polytope only (dim(MKP)=n). Standard linear relaxation amounts to turn binary constraints on x into box constraints $x \in [0,1]^n$:

$$\begin{aligned} & \max\ \langle c, x\rangle \\ \text{s.t.}\ & Ax \leq b \\ & x \in [0,1]^n \end{aligned} \tag{LRMKP}$$

However, it introduces fractional points useless for original problem so that we keep equivalence only if we further require feasible points to be outside an hypersphere close enough to the unit hypersphere centered at $\frac{e}{2}$ where e stands for the all ones vector.

$$\begin{aligned} & \max\ \langle c, x\rangle \\ \text{s.t.}\ & Ax \leq b \\ & x \in [0,1]^n \\ & g_\delta(x) \geq 0 \end{aligned} \tag{RCMKP}$$

where $g_\delta(x) = \langle x - \frac{e}{2}, x - \frac{e}{2}\rangle - \frac{n}{4} + \delta$ and we will name it geodesic preferably to level set to leave opened possibly change in δ.

Clearly, for $\delta = 0$, both problems are equivalent (MKP) $\equiv_0$ (RCMKP); yet, it is well known [Tse96] that for single knapsack problem (SKP), exists δ_s such that for all $\delta \leq \delta_s$, (SKP) $\equiv_\delta$ (RCSKP). It carries over (MKP) by taking minimum δ_s over all knapsack constraints; therefore, for δ small enough: (MKP) $\equiv_\delta$ (RCMKP). The latter form is known as reverse convex optimization.

The main purpose of this paper is to compare a global optimization approach to (MKP) with heuristics performance with respect to speed and quality of solution retrieved. It is known from solving high dimensional nonconvex problems that pure cutting plane methods may fail ([GJ91]) and B&B is impractical, due to a large duality gap. On the other hand, $\Re$-strategy ([ST98]) does not help much because we do not know how to generate all geodesic points. Therefore we propose to combine both a cutting plane method and $\Re$-strategy with a random generation of geodesic points where the number of points is limited by a tabu list to prevent re-examination of the same geodesic area. Experiments show that we end up with a small duality gap permitting a subsequent Branch&Bound approach for reasonable sized instances.

In sections 2, 3 we use known results from global optimization [KST99] and tailor them for (MKP); in section 4, we introduce some equivalence relation to reduce optimality condition checking.

Finally, in section 5, we give computational results to show how favorably this algorithm compares with *standard* heuristics [CB98, DV91, Dre88, FP96, HF98, KBH94].

2. OPTIMALITY CONDITIONS

Let a reverse convex problem be given by

$$\begin{aligned} &\min \quad f(x) \\ &\text{s.t.} \quad x \in \mathbb{F} \subset \mathbb{R}^n, g(x) \geq 0 \end{aligned} \tag{RCP}$$

under $f, g, \mathbb{F}$ being convex. Constraint $g(x) \geq 0$ is said reverse convex.

DEFINITION 2.1. [actual reverse convex problem] (RCP) is an actual reverse convex problem if there exists $v \in \mathbb{R}^n$ with $g(v) < 0$; otherwise reverse convex constraint is trivially fulfilled.

DEFINITION 2.2. [reverse convex problem *regularity*] (RCP) is *regular* whenever for all $y \in \mathbb{F} : g(y) = 0,\quad$ there exists $x \in \mathbb{F} \ : \ \langle \nabla g(y), x - y \rangle \ > 0$.

THEOREM 2.1. *(Strekalovskii [Str93]) For (RCP) being an actual and regular reverse convex problem, let z be a feasible point*

$$\begin{aligned} &\textit{if for all} \quad y : g(y) = 0 \ \textit{and for all} \ x \in \mathbb{F} : f(x) \leq f(z), \langle \nabla g(y), x - y \rangle \leq 0 \ \textit{holds} \\ &\textit{then} \quad z \ \textit{is a global solution to (RCP)} \end{aligned}$$

In order to check the sufficient optimality condition, one has to solve a collection of *linearized* problems

$$\text{for all } y : g(y) = 0, \max \langle \nabla g(y), x \rangle \text{ s.t. } x \in \mathbb{F}, f(x) \leq f(z) \tag{1}$$

and then check whether

$$\langle \nabla g(y), u(y) - y \rangle \leq 0 \tag{2}$$

is satisfied for all $y : g(y) = 0$ at $u(y)$ solution of linearized problem.

Claim. Let $\mathbb{F} = [0,1]^n \cap Ax \leq b$ be intersection of box constraints and knapsack constraints, then (RCMKP) is both an actual and regular reverse convex problem. Therefore the condition of the theorem 2.1 is a sufficient for global optimality in (RCMKP) under δ small enough.

Proof. $\textit{arg}\min \ g_\delta(x) = \frac{e}{2}$ guarantees first condition ($g_\delta(\frac{e}{2}) = \delta - \frac{n}{4}$) while $\delta > 0$ implies existence of required points in curved corners. ■

In the case of (RCMKP) we have sufficient optimality condition of being z global solution:

$$\begin{gathered}\text{for all} \quad y : g_\delta(y) = 0, \\ \langle \nabla g_\delta(y), x - y \rangle \leq 0, \\ \text{for all } x \in [0,1]^n, Ax \leq b, -\langle c, x \rangle \leq -\langle c, z \rangle. \end{gathered}$$

Since it is intractable to check inequality (2) for all $y : g_\delta(y) = 0$, it suggests to sample geodesic by *shooting* in random directions as long as condition on geodesic point (met along direction) is satisfied; otherwise, we found a better point and continues from this **percolated** point. Stopping criterion involves maximum number of shots or small gap between upper bound and $\langle c, z \rangle$ in (RCMKP). We refine this random shooting by first looking at easy (analytically found) geodesic points along box and knapsack constraints and then at random direction as explained later in section 5.

3. CUTTING PLANE

From global optimization [HT90], we adapt standard retrieval of a cutting plane to (MKP). Let us assume that $u \in \mathbb{R}^n$ solves problem (LRMKP). Obviously, if $u \in \{0,1\}^n$ then it solves (MKP) as well, else we have to cut this fractional point.

Let $P = \{x \mid \quad Ax \leq b, \quad x \in [0,1]^n\}$ and $u = \arg\max_{x \in P} \langle c, x \rangle$ s.t. $g_\delta(u) < 0$ be given. Let $A(u)$ denote the matrix of active constraints at u. Then from $A^{-1}(u)$ (inverse of $A(u)$) and kronecker product $\otimes$, one gets

$$Z = u \otimes e^\top + A^{-1}(u)\alpha$$

and

$$Y = u \otimes e^\top + A^{-1}(u)\beta$$

for some $\alpha \in R^n_-$ such that every column vector z_i of Z belongs to P and some $\beta \in R^n_-$ such that every column vector y_i of Y fulfills $g_\delta(y_i) = 0$.

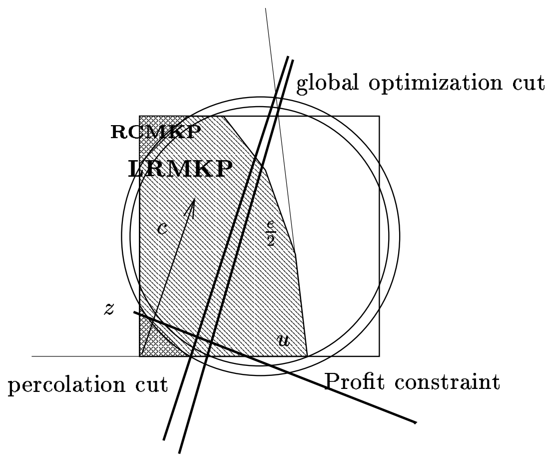

FIG. 1. global and percolation cuts

Claim. Let Y be defined as above and d be unique solution of system of linear equations

$$Yd = ne.$$

then hyperplane $\langle d, x \rangle = n$ (see Fig. 1 *percolation cut*) cuts only fractional points together with u.

Proof. Due to convexity of function $g_\delta(\cdot)$ and

$$g_\delta(y_i) = 0(i = 1, 2, ..., n), \quad g_\delta(u) < 0$$

we have $g_\delta(x) \leq 0$ for all x such that

$$x = u(1 - \sum_{i=1}^{n} \gamma_i) + \sum_{i=1}^{n} \gamma_i y_i, \quad \sum_{i=1}^{n} \gamma_i \leq 1, \ \gamma_i \geq 0(i = 1, 2, ..., n).$$

■

It is worthwhile to notice that any constant in right hand side is convenient; however n happens to be numerically stable in our experiments whereas 1 introduces highly unstable behavior.

4. GEODESIC PARTITIONING

In this section, we specialize inverse LP problems to partition geodesic into equivalence classes w.r.t. percolation phase. For any LP $\max_x \langle c, x\rangle$ on set $P = \{x \mid Ax \leq b\}$, and feasible solution $\hat{x} \in P$, we associate a so-called *inverse* LP: find $\hat{c}, \hat{b}$ such that $\hat{x} = \arg\max_x \langle \hat{c}, x\rangle$ on set $\hat{P} = \{x \mid Ax \leq \hat{b}\}$.

LEMMA 4.1. *Let u be a vertex of polytope $P = \{x \mid Ax \leq b\}$ and $A(u)$ be the set of active constraints at u; then*
for all $\alpha \geq 0, \langle \alpha, e\rangle = 1$, $u \in \arg\max\{\langle A(u)\alpha, x\rangle \mid Ax \leq b\}$ holds.

Proof. noting $d = A(u)\alpha$, we have for all $x \in P, \langle d, x\rangle \leq \langle d, u\rangle$ since $Au \leq b$ and $\alpha \geq 0$. ■

Remark. Percolation or global optimization cuts in section 3 come from inverse LP where the actual depth of hyperplanes is related to corresponding geodesics.

COROLLARY 4.1. *For a* **full dimensional** *polytope $P = \{x \mid Ax \leq b, x \in [0,1]^n\}$ and reverse convex problem $\min_x f(x)$ s.t. $\{x \in P, g(x) \geq 0\}$, let u be a solution of linear relaxation $\min_{x \in P_{LR}} f(x)$ on relaxed polytope $P_{LR} = \{x \mid Ax \leq b\}$, then u is representative for all $x \in P$ convex combination of active constraints at u, $x = A(u)\alpha$ for $\alpha \in (0,1)^n$.*

Proof. since polytope is full dimensional, $\arg\min\{\langle A(u)\alpha, x\rangle \mid Ax \leq b\}$ is unique. ■

Since (MKP) is full dimensional, it suggests to introduce some tabu list along random generation of geodesic points; let $y, g_\delta(y) = 0$ a geodesic point generated by random direction d, then d is tabu if it is convex combination of active direction $A(u)$ for some u solution of linear relaxation (LRMKP). To be more precise, $y = \frac{e}{2} + \frac{d}{\|d\|}\frac{n-4\delta}{2}$ for convex combination d at u leads to already examined geodesic area.

5. COMPUTATIONAL RESULTS

5.1. a *percolation* algorithm

ALGORITHM 1 (MKP($\max\langle c,x\rangle, Ax \le b, x \in \{0,1\}^n$)).

repeat
u:=simplex($\max\langle c,x\rangle, Ax \le b, x \in [0,1]^n$)); /* (LRMKKP) */
if ($z = [u]$ or $z = \lfloor u \rfloor$) better solution
then add $-\langle c,x\rangle \le -\langle c,z\rangle$; continue **endif;** /* trivial cutting */
until SEPARATE($u, A(u)$) $\neq \emptyset$; /* cutting plane */

where $A(u)$ stands for active constraints at u while $[u]$ (resp. $\lfloor u \rfloor$) denotes rounding (resp. taking the floor) componentwise.

ALGORITHM 2 (SEPARATE($u, A(u)$)).

URAY:= $A(u)^{-1}$; /* extreme rays of active cone */
U:= $\emptyset$; /* geodesic points from active cone */
V:= $\emptyset$; /* feasible adjacentvertices along extreme rays */
foreach ray **in** URAY **do**
find y s.t. $\{(y-U) \wedge$ ray $= 0, g_\delta(y) = 0\}$
if ($z = [y]$ or $z = \lfloor y \rfloor$) better solution
then
u:=simplex($\max\langle c,x\rangle, Ax \le b, -\langle c,x\rangle \le -\langle c,z\rangle, x \in [0,1]^n$));
return SEPARATE($u, A(u)$); /* trivial cutting */
else
U:= U$\cup y$;
V:= V$\cup z$; s.t. $\{(z-u) \wedge (y-u) = 0, z \in$ (LRMKP)$\}$
endif;
endfor /* U is cutting plane by above */
l:=arg min $\langle c,v\rangle$ over $v \in$ V; /* lower bound simplex */
LRAY:= $A(l)^{-1}$; /* extreme rays at lower bound cone */
L:= $\emptyset$; /* geodesic points from lower bound cone */
foreach ray **in** LRAY **do**
find y s.t. $\{y \wedge$ ray $= 0, g_\delta(y) = 0\}$
if ($z = [y]$ or $z = \lfloor y \rfloor$) better solution
then
u:=simplex($\max\langle c,x\rangle, Ax \le b, -\langle c,x\rangle \le -\langle c,z\rangle, x \in [0,1]^n$));
return SEPARATE($u, A(u)$); /* trivial cutting */
else
L:= L $\cup y$;

    endif;
endfor *$/*$ L is cutting plane by below $*/$*
CUTS:= L $\cup$ U;
if CUTS efficient
then return CUTS;
else *$/*$ cuts too close from $u \longrightarrow$ interrupt cutting $*/$*
    v:= PERCOLATE(u);
    if $(v \neq u)$
    then return SEPARATE$(u, A(u))$ **endif;**
endif;
return $\emptyset$;

ALGORITHM 3 (PERCOLATE(u)).

z:= $u(\delta)$ *$/*$ geodesic point close to u corner $*/$*
lp:= $\{\max\langle\nabla g_\delta(y), x\rangle, Ax \leq b, -\langle c, x\rangle \leq -\langle c, z\rangle, x \in [0,1]^n\}$
foreach $\langle e_i, x\rangle = 0$ or $\langle e_i, x\rangle = 1$ **do**
    v := sibling box constraint point *$/*$ $2n$ such points $*/$*
    if (v) better point **then return** v **endif;**
endfor
foreach $A(u) \cup \{-\langle c, x\rangle \leq -\langle c, z\rangle\}$ active constraints **do**
    find v s.t. $\{\| u - v \|$ maximum, $g_\delta(v) = 0\}$
    if (v) better point **then return** $v(\delta)$ **endif;**
endfor
for $l = 1$ **to** maxiteration **do**
    generate random direction d not in tabu list;
    v:= farthest feasible point from z along d;
    find $v1, v2$ s.t. $\{g_\delta(v1) = 0, g_\delta(v2) = 0,\ (v2 - v1\wedge)d_{\perp\langle a,x\rangle=\langle a,v\rangle} = 0\}$
    if $(v_i, i = 1, 2\ )$ better point **then return** $v_i(\delta)$ **endif;**
endfor
return u; *$/*$ no better point found $*/$*

where $u(\delta), v(\delta)$ mean that a feasible geodesic point is retrieved from lower bound l corner along max profit component (in order to increase current lower bound $\langle c, l\rangle$ by a small amount depending on δ) and where random direction is projected onto active constraint at $v(d_{\perp\langle a,x\rangle=\langle a,v\rangle})$ before retrieving $v1, v2$ geodesic farthest points from v.

5.2. remarks

Previous algorithm belongs to cutting class of algorithms since it alternates cutting phase with percolation phase; however percolation phase should not be compared with primal improvement heuristics since it relies on global optimality sufficient condition instead of local search improvement.

Let notice that better point in percolate algorithm is related to problem linearized from (RCMKP) instead of profit function in separation phase of outermost algorithm loop (LRMKP) while constraint remain the same.

In lines 8 and 22 of SEPARATE algorithm, profit constraint seems redundant but helps as a stopping criterion when duality gap is small (this early stop was forecasted as a generic feature of global optimization, compared to linear relaxation, in section 2).

In line 17 of SEPARATE algorithm, we select only one vertex from simplex defined by apex u the current best fractional solution to (LRMKP) and n feasible adjacent vertices to u. In our first attempts, we included all simplicial vertices cuts at the cost on $n+1$ $n \times n$ matrix inversions; then we restricted to *upper* and *lower* cuts only and save one order of magnitude in computation time without losing accuracy in results. A possible interpretation for no loss of accuracy lay on the compromise between better approximation of polytope ($n+1$ cuts) and less numerous explicit fractional points introduced by 2 cuts.

Another consequence of previous remark on tradeoff between accuracy of polytope approximation and number of explicit fractional points is that we observed better solutions whenever we selected only *knapsack* cuts (normal in positive orthant) instead of deep cuts with possibly negative direction. However, this effect is blurred by the fact that *shooting* in knapsack supporting hyperplane is likely to give *better* geodesic points.

In line 12 of PERCOLATE algorithm, we used a tabu list (see section 4) remembering last n fractional points as a vector of booleans tagging which constraints are active in this point; thus, it requires no more than $n/32$ words on most computers for each entry. We did not further study tabu list size on algorithm behavior. Algorithm's naming comes from *percolation* of profit constraint through equivalence classes on geodesic i.e. sieving along with objective improvement.

We also introduce a refinement in random loop by computing an average random direction and use a convex combination instead $(1-\alpha)$ times average direction plus α times true random direction; starting with $\alpha = 1$ (pure **diversification**), each maxiteration loops, α is halved upto 0.05 so that **intensification** becomes gradually more influential.

All experiments were done under ABACUS[1] and CPLEX. [2]

[1]ABACUS a software framework for the development of optimization algorithms distributed by OREAS http://www.oreas.de/frames.html.

[2]CPLEX is a registered trademark of ILOG Copyright ©1997 http://www.ilog.com.

5.3. intensification/diversification results

Our experiments are done on mknapcb1.txt, a collection of test data sets from http://mscmga.ms.ic.ac.uk/jeb/orlib/mknapinfo.html.

Maximum number of iteration loops before increasing intensification is set to 100. We report the following indicators in table 1:

- first sol. = value obtained after first cutting round,
- final sol. = value at stopping criterion,
- first gap = (first sol.-best known)/max(first sol.,best known) in percentage,
- final gap = (final sol.-best known)/max(final sol.,best known) in percentage,
- dual. gap = (final sol.-upper bound)/upper bound in percentage,
- time is from DEC PWS500 workstation,
- #percolation = number of geodesic points examined (on box, knapsack or randomly generated)

Comparison between first gap and final gap shows that both cutting and percolation are efficient. While algorithm affords to interleave cutting and percolation, no improvement of upper bound was observed after first cutting round.

5.4. Branch and Cut results

Duality gap in previous experiment suggested to try a Branch and Cut algorithm instead of a pure percolation algorithm. Intensification is completely discarded and branching is done on close half expensive variable after each round of percolation. Our experiments are done on mknap2.txt, a collection of test data sets from http://mscmga.ms.ic.ac.uk/jeb/orlib/mknapinfo.html since optimal solution are known.

Maximum number of iteration loops for percolation is now set to n. We keep meaningful indicators from previous test and add the following for table 2:

- Problem = name and size as $(m \times n)$
- first(sec)= solution after first cutting round and time in seconds on DEC PWS500 workstation,
- opt. sol. = optimal solution,
- #sub = number of subproblems in B&B tree,
- #level = highest level in B&B tree,
- #percolation = total number of geodesic points generated

TABLE 1

percolation with intensification on mknapcb1.txt: $m = 5, n = 100$

#Pb.	first sol.	final sol.	best known	first gap	final gap	upper bound	dual. gap	time(sec)	#percolation
0	23061	23728	24381	-5.41%	-2.67%	24584.3	-3.48%	1955	1200
1	23818	23818	24274	-1.87%	-1.87%	24536	-2.92%	13697	1200
2	22102	23257	23551	-6.15%	-1.24%	23893.3	-2.66%	1960	1200
3	20943	22404	23534	-11%	-4.8%	23717.7	-5.53%	5941	1200
4	23281	23281	23991	-2.95%	-2.95%	24215.8	-3.86%	1964	1200
5	24339	24339	24613	-1.11%	-1.11%	24872.9	-2.14%	4904	1200
6	23037	25351	25591	-9.98%	-0.93%	25793.4	-1.71%	993	1200
7	21543	22855	23410	-7.97%	-2.37%	23651.9	-3.36%	3917	1200
8	21814	23127	24216	-9.91%	-4.49%	24445.6	-5.39%	990	1200
9	22736	23826	24411	-6.86%	-2.39%	24635.7	-3.28%	994	1201
10	40758	42705	42757	-4.67%	-0.12%	42931.9	-0.52%	1961	1201
11	40070	42246	42545	-5.81%	-0.7%	42706.7	-1.07%	994	1200
12	40875	41755	41968	-2.6%	-0.5%	42164.1	-0.97%	1959	1200
13	43512	44680	45090	-3.49%	-0.9%	45341	-1.45%	1964	1200
14	41229	41764	42218	-2.34%	-1.07%	42427.1	-1.56%	1978	1200
15	42271	42681	42927	-1.52%	-0.57%	43081	-0.92%	1965	1200
16	41980	41980	42009	-0.06%	-0.06%	42185.6	-0.48%	2933	1200
17	43563	44704	45020	-3.23%	-0.7%	45261.5	-1.23%	2933	1200
18	42935	43122	43441	-1.16%	-0.73%	43566.3	-1.01%	2938	1201
19	44511	44511	44554	-0.09%	-0.09%	44795.3	-0.63%	1973	1200
20	58466	59642	59822	-2.26%	-0.3%	60016.6	-0.62%	993	1200
21	60520	61805	62081	-2.51%	-0.44%	62224.6	-0.67%	1965	1200
22	58363	59297	59802	-2.4%	-0.84%	59959.6	-1.1%	6863	1200
23	60151	60151	60479	-0.54%	-0.54%	60642.4	-0.81%	1965	1200
24	59747	60991	61091	-2.19%	-0.16%	61330.1	-0.55%	1967	1200
25	57915	58920	58959	-1.77%	-0.06%	59161.9	-0.4%	991	1200
26	61383	61383	61538	-0.25%	-0.25%	61692.5	-0.5%	3894	1200
27	61184	61366	61520	-0.54%	-0.25%	61718.9	-0.57%	2937	1200
28	57638	59298	59453	-3.05%	-0.26%	59610.5	-0.52%	10756	1200
29	59650	59650	59965	-0.52%	-0.52%	60230.5	-0.96%	2938	1200

TABLE 2

Branch and Cut under percolation on mknap2.txt

Problem	first(sec)	opt. sol.	first gap	upper bound	dual. gap	time(sec)	#level	#sub	#percolation
SENTO1.DAT(30×60)	7761(27)	7772	-0.14%	7780.37	-0.1%	2373	40	631	75600
SENTO2.DAT(30×60)	8716(41)	8722	-0.06%	8723.85	-0.02%	5828	35	1363	163440
WEING1.DAT(2×28)	141278(1)	141278	0%	141332	-0.03%	33	17	91	5043
WEING2.DAT(2×28)	130883(1)	130883	0%	130929	-0.03%	16	17	49	2691
WEING3.DAT(2×28)	95677(1)	95677	0%	95691.2	-0.01%	27	24	81	4480
WEING4.DAT(2×28)	115687(2)	119337	-3.05%	119370	-0.02%	44	16	119	5936
WEING5.DAT(2×28)	98016(1)	98796	-0.78%	98840	-0.04%	13	19	43	1568
WEING6.DAT(2×28)	130233(1)	130623	-0.29%	130634	0%	27	12	79	4386
WEING7.DAT(2×105)	1095360(33)	1095445	0%	1095460	0%	1002	16	79	16419
WEING8.DAT(2×105)	609580(56)	624319	-2.36%	628774	-0.7%	10086	77	829	173880
WEISH01.DAT(5×30)	4554(4)	4554	0%	4561.63	-0.16%	82	21	117	6960
WEISH02.DAT(5×30)	4506(1)	4536	-0.66%	4543.4	-0.16%	22	10	33	1693
WEISH03.DAT(5×30)	4506(1)	4115	-0.66%	4137.99	-0.55%	38	25	79	4680
WEISH04.DAT(5×30)	4561(1)	4561	0%	4570.81	-0.21%	21	13	31	1800
WEISH05.DAT(5×30)	4514(1)	4514	0%	4530.78	-0.37%	14	15	29	1680
WEISH06.DAT(5×40)	5557(3)	5557	0%	5571.9	-0.26%	119	31	111	8804
WEISH07.DAT(5×40)	5567(6)	5567	0%	5567.52	0%	210	23	117	9280
WEISH08.DAT(5×40)	5603(7)	5605	-0.03%	5605.79	-0.01%	110	21	93	7364
WEISH09.DAT(5×40)	5246(2)	5246	0%	5247.13	-0.02%	39	22	45	3520
WEISH10.DAT(5×50)	6339(7)	6339	0%	6370.79	-0.49%	493	34	289	28800
WEISH11.DAT(5×50)	5643(3)	5643	0%	5644	-0.01%	178	32	109	10800
WEISH12.DAT(5×50)	6339(3)	6339	0%	6357.26	-0.28%	222	31	133	13200
WEISH13.DAT(5×50)	6159(7)	6159	0%	6159.65	-0.01%	171	35	103	10200
WEISH14.DAT(5×60)	6954(6)	6954	0%	6955.29	-0.01%	298	35	119	14160
WEISH15.DAT(5×60)	7486(6)	7486	0%	7486.82	-0.01%	215	22	91	0
WEISH16.DAT(5×60)	7273(8)	7289	-0.21%	7289.36	0%	382	38	153	18240
WEISH17.DAT(5×60)	8624(6)	8633	-0.1%	8635.83	-0.03%	109	13	37	4321
WEISH18.DAT(5×70)	9565(30)	9580	0%	9585.67	-0.05%	1528	33	319	44587
WEISH20.DAT(5×70)	9441(10)	9450	-0.09%	9452.03	-0.02%	598	37	139	19320
WEISH21.DAT(5×70)	9040(11)	9074	-0.37%	9110.5	-0.4%	1032	51	247	34440
WEISH22.DAT(5×80)	8914(22)	8947	-0.36%	9004.18	-0.63%	1610	55	261	41664
WEISH23.DAT(5×80)	8341(32)	8344	-0.03%	8344.58	0%	1191	45	193	30720
WEISH24.DAT(5×80)	10192(20)	10220	-0.27%	10223	-0.02%	1222	46	197	31360
WEISH25.DAT(5×80)	9875(28)	9939	-0.64%	9952.1	-0.13%	3552	48	611	39995
WEISH26.DAT(5×90)	9538(24)	9584	-0.47%	9587.09	-0.03%	3340	57	387	69480
WEISH27.DAT(5×90)	9819(28)	9819	0%	9849.67	-0.31%	1554	54	179	32040
WEISH28.DAT(5×90)	9492(16)	9492	0%	9492.57	0%	967	54	113	20160
WEISH29.DAT(5×90)	9410(20)	9410	0%	9420	-0.1%	1177	51	137	24479
WEISH30.DAT(5×90)	11174(32)	11191	-0.15%	11191.6	0%	240	11	29	360
PB1.DAT(4×27)	3034(3)	3090	-1.81%	3100.21	-0.32%	15	13	35	972
PB2.DAT(4×34)	2974(13)	3186	-6.65%	3190.27	-0.13%	267	16	305	9299
PB4.DAT(2×29)	74400(2)	95168	-21.82%	99622.7	-4.47%	107	22	309	17864
PB5.DAT(10×20)	2139(3)	2139	0%	2149.88	-0.5%	1668	21	707	28080
PB6.DAT(30×40)	765(24)	776	-1.41%	778.517	-0.32%	990	34	443	35360
PB7.DAT(30×37)	1029(11)	1035	-0.57%	1042.53	-0.72%	1880	27	1237	91464
HP1.DAT(4×28)	3362(4)	3418	-1.63%	3428.21	-0.29%	18	11	33	1008

6. CONCLUDING REMARKS

Fisrt of all, we addressed (MKP) from a global optimization viewpoint; more precisely we turn it into a reverse convex problem. However, further shed must be put on the *convexification* process at the theoretical level, parameters have to be tuned (#cutting planes vs #percolation points) and full hybridation with known heuristics should be tailored to yield a fast algorithm dedicated to (MKP) application.

6.1. Dual Lagrangian

Percolation algorithm proposed in this paper, only relies on *standard* cut; however, there is an outside counterpart of inside cut approach. From box constraints $0 \leq x \leq 1$, we could introduce redundant quadratic constraints
$\langle x, x-e \rangle \leq 0$ into (RCMKP) and then take the Lagrangian

$$\begin{array}{rl} \min_x & \mathcal{L}(x, \lambda, p) \\ \text{s.t.} & Ax \leq b, \\ & x \in [0,1]^n, \\ & g_\delta(x) \geq 0, \\ & \lambda \in \mathbb{R}_+,\ \rho \in \mathbb{R}^m_+, \end{array} \qquad \text{(LRCMKP)}$$

where

$$\begin{aligned} \mathcal{L}(x, \lambda, \rho) &= -\langle c, x \rangle + \lambda \langle x, x-e \rangle + \langle \rho, Ax - b \rangle \\ &= \lambda \parallel x - x(\lambda, \rho) \parallel^2 - [\lambda \parallel x(\lambda, \rho) \parallel^2 + \langle \rho, b \rangle] \end{aligned}$$

with

$$x(\lambda, \rho) = \frac{e}{2} + \frac{1}{2\lambda}(c - A^\top \rho).$$

The Lebesgue set $\{x \mid \mathcal{L}(x, \lambda, \rho) \leq \mathcal{L}(z, \lambda, \rho)\} = \{x \mid \parallel x - x(\lambda, \rho) \parallel \leq r_{\mathcal{L}}\}$ which contains minimum of (LRCMKP) is the ball with radius $r_{\mathcal{L}} = \parallel z - x(\lambda, \rho) \parallel$ and center $x(\lambda, \rho)$. On the other hand we have reverse convex constraint $g_\delta(x) \geq 0$ which removes interior of the ball with radius $r_g = (\frac{n}{4} - \delta)^{\frac{1}{2}}$ and center $\frac{e}{2}$. Then we define Lagrangian cut as following.

Claim. For all $v \in \{x \mid \parallel x - x(\lambda, \rho) \parallel^2 \leq r^2_{\mathcal{L}}\} \setminus \{x \mid \parallel x - \frac{e}{2} \parallel^2 < r^2_g\}$:

$$\langle c - A^\top \rho, v \rangle \geq \lambda (\parallel \frac{1}{2\lambda}(c - A^\top \rho) \parallel^2 - r^2_{\mathcal{L}} + r^2_g)$$

Remark. Lagrangian cut in the unknowns λ, ρ can be, for example, fully determined from the solution of the following system of equations:

$$\begin{aligned} Ax(\lambda, \rho) &= b \\ \langle c, x(\lambda, \rho) \rangle &= \langle c, z \rangle. \end{aligned}$$

let $\gamma = \frac{1}{2\lambda}, \sigma = -\frac{1}{2\lambda}\rho$ replace unknowns λ, ρ, then it leads to $(m+1) \times (m+1)$ linear system:

$$\begin{aligned} \gamma Ac + AA^\top \sigma &= b - \frac{1}{2}Ae \\ \gamma \parallel c \parallel^2 + \langle c, A\sigma \rangle &= \langle c, z - \frac{e}{2} \rangle \end{aligned} \tag{3}$$

whose solution is unique from definite positiveness of matrix $AA^\top$ and $\parallel c \parallel^2 > 0$. In figure 2, both cuts are drawn and we may expect that for some λ, ρ lagrangian cut may outperform perculation cut (see section 3).

We could notice, furthermore, that standard Lagrangian rule in addition to $\partial \mathcal{L}(x, \lambda, p)/\partial x = 0$ requires $\partial \mathcal{L}(x, \lambda, \rho)/\partial \rho \geq 0$ and $\partial \mathcal{L}(x, \lambda, \rho)/\partial \lambda \geq 0$; the latter yields $g_\delta \geq 0$, which points out how Lagrangian relaxation turns a minimization problem into a reverse convex problem (at first sight (RCMKP) derivation appears tricky but its flavor comes from Lagrangian dual on $\langle x, x - e \rangle$).

6.2. computational tradeoffs

For sake of comparison results, we consider as starting feasible solution one obtained by pure cutting plane initial phase; of course, any primal heuristics giving a *good* starting point in a very smal amount of time could be plugged in above algorithm to improve running time. We guess that it will improve numerical stability as well, since fewer cutting planes would be introduced whenever profit constraint is good enough.

6.3. further research

Percolation approach would extend to any combinatorial problems in binary variables provided the following points have been solved for non full dimensional case

- geodesic partitioning,
- cutting plane lifting.

While lifting sounds harder than partitioning, we guess that it could be achieved; then, together with a deep lagrangian cut, very hard problems like QAP would be worthwhile to test through this global optimization approach.

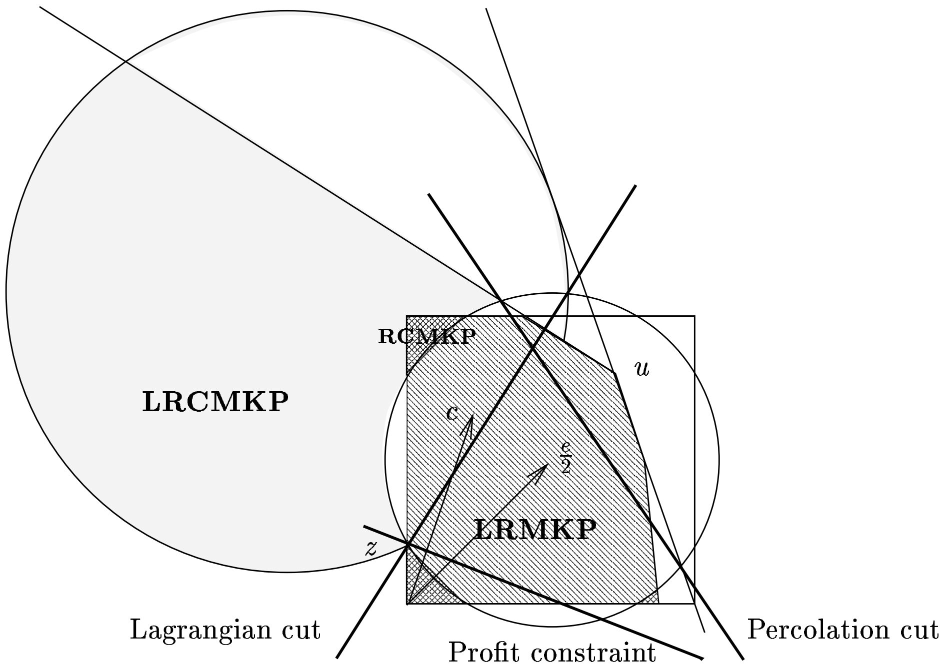

FIG. 2. multi knapsack relaxations and cuts

REFERENCES

CB98. P. C. Chu and J. E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.

Dre88. A. Drexel. A simulated annealing approach to the multiconstraint zero-one knapsack problem. *Computing*, 40:1–8, 1988.

DV91. F. Dammeyer and S. Voss. Dynamic tabu list management using the reverse elimination method. *Annals of Operations Research*, 1991.

FP96. A. Fréville and G. Plateau. The 0-1 bidimensional knapsack problem: towards an efficient high level primitive tool. *Journal of Heuristics*, 2, 1996.

GJ91. T. R. Gurlitz and S. E. Jacobsen. On the use of cuts in reverse convex programs. *Annals of Operations Research*, 68(2):257–274, February 1991.

HF98. S. Hanafi and A. Fréville. An efficient tabu search approach for the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*, (106):659–675, 1998.

HT90. R. Horst and H. Tuy. *Global Optimization.* Springer-Verlag, 1990.

KBH94. S. Khuri, T. Baeck, and J. Heitkoetter. The zero/one multiple knapsack problem and genetic algorithms. In *ACM Symposium on Applied Computing, SAC'94*, pages 188–193, Phoenix, Arizona, 1994.

KST99. A. A. Kuznetsova, A. S. Strekalovskii, and I. Tsevendorj. An approach to the solution of integer optimization problems. *Computational Mathematics and Mathematical Physics*, 39(1):6–13, 1999.

ST98. A. S. Strekalovskii and I. Tsevendorj. Testing the R-strategy for a reverse convex problem. *Journal of Global Optimization*, 13:61–74, 1998.

Str93. A. S. Strekalovskii. Extremal problems on complements of convex sets. *Kibernetika i Sistemnii. Analiz*, (1):113–126, 1993. translated in Cybernetic Systems Analysis(29) (1993)1,pp 88–100.

Tse96. I. Tsevendorj. *A Problem About Searching a Global Solution in Reverse Convex Problem.* PhD thesis, Irkutsk State University (Russia), 1996.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399