



حل مساله کوله پشتی چندبعدی با استفاده از اتماتاهای یادگیر

محمد رضا میدی

عضو هیات علمی دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

mmeybodi@aut.ac.ir

سمیرا نوferستی

عضو هیات علمی دانشگاه سیستان و بلوچستان

دانشکده مهندسی شهید نیکبخت، گروه فناوری اطلاعات

snoferesti@ece.usb.ac.ir

که x_j متغیر تصمیم‌گیری متناظر با کالای j است. در صورتی که کالای j از انتخاب شود، x_j مقدار ۱ و در غیر این صورت مقدار صفر را اختیار می‌کند. متغیرهای p_j , w_{ij} و b_i نمی‌توانند مقادیر منفی اختیار کنند.

به دلیل اهمیت مساله کوله‌پشتی تاکنون الگوریتمهای متعددی برای حل آن گزارش شده است. این الگوریتمها را می‌توان به دو گروه کلی تقسیم کرد: الگوریتمهای دقیق و الگوریتمهای تقریبی. با توجه به اینکه مساله کوله‌پشتی یک مساله NP_Complete می‌باشد الگوریتمهای دقیق که معمولاً از روش‌های برش و انشعاب^۲ یا روش‌های ترکیبی با برنامه‌نویسی پویا^۳ استفاده می‌کنند [۴،۵]. در بدترین حالت دارای پیچیدگی نمایی هستند و برای استفاده در کاربردهای عملی مناسب نمی‌باشند. به همین دلیل الگوریتمهای تقریبی متعددی برای حل مساله کوله‌پشتی گزارش شده است. بسیاری از تحقیقات انجام شده به حل مساله کوله‌پشتی تک بعدی^۴ ($m=1$) پرداخته‌اند [۵،۶]. تلاش‌های زیادی نیز در جهت حل تقریبی مساله کوله‌پشتی چندبعدی انجام گرفته است. از جمله الگوریتمهای تقریبی برای حل مساله کوله‌پشتی چندبعدی، الگوریتمهای تکرارشونده^۵ هستند. در این روشها رسیدن به یک پاسخ بهینه تضمین نمی‌شود اما در اغلب موارد جوابهای تقریبی قابل قبولی تولید می‌کنند. از جمله الگوریتمهای تکرارشونده می‌توان به الگوریتم ژنتیکی [۷،۸،۹] و الگوریتم کلونی مورچه‌ها [۱۰،۱۱] اشاره کرد.

در این مقاله یک الگوریتم تکرارشونده مبتنی بر اتماتاهای یادگیر برای حل مساله کوله‌پشتی چندبعدی پیشنهاد می‌شود. در این الگوریتم مساله کوله‌پشتی با یک گراف کامل مدل می‌شود که هر گره از گراف متناظر با یکی از کالاهاست. هر گره از گراف به یک اتماتای یادگیر با دو عمل مجهز است که انتخاب یا عدم انتخاب کالای متناظر برای قرار گرفتن در کوله‌پشتی را مشخص می‌کند. در هر تکرار از الگوریتم

چکیده: در این مقاله یک الگوریتم تکرارشونده مبتنی بر اتماتاهای یادگیر برای حل مساله کوله‌پشتی چندبعدی پیشنهاد می‌شود. در این الگوریتم، مساله کوله‌پشتی با یک گراف کامل مدل می‌شود که هر گره از گراف متناظر با یکی از کالاهاست. هر گره از گراف به یک اتماتای یادگیر مجهز است که انتخاب یا عدم انتخاب کالای متناظر با گره برای قرار گرفتن در کوله‌پشتی را مشخص می‌کند. نتایج شبیه‌سازی‌ها نشان داده است که الگوریتم پیشنهادی در مقایسه با الگوریتمهای موجود از کارایی بالاتری برخوردار است. نتایج شبیه‌سازی‌ها همچنین نشان داده است که الگوریتم پیشنهادی برای مسائل با اندازه‌های بزرگ دارای سرعت همگرایی بالایی می‌باشد.

واژه‌های کلیدی: مساله کوله‌پشتی چندبعدی، اتماتاهای یادگیر، مسایل مشکل

۱ - مقدمه

مساله کوله‌پشتی چندبعدی^۱ یکی از مسایل بهینه‌سازی ترکیبی است که کاربردهای فراوانی از قبیل تخصیص منابع، برنامه‌ریزی بودجه، تخصیص سهام و بارگیری محموله‌ها دارد. مساله کوله‌پشتی چندبعدی شامل m کوله‌پشتی با ظرفیتهای b_1, b_2, \dots, b_m و n کالا است. کالای i دارای ارزش p_i می‌باشد و وزن w_{ij} از کوله‌پشتی j را اشغال می‌کند. هدف پر کردن کوله‌پشتی‌ها با زیرمجموعه‌ای از کالاهای این مجموعه است که بیشترین سود حاصل شود و مجموع وزن کالاهای یک کوله‌پشتی از ظرفیت آن تجاوز نکند. یک کالا یا در همه کوله‌پشتی‌ها قرار می‌گیرد یا برای هیچیک از کوله‌پشتی‌ها انتخاب نمی‌شود. به طور دقیقترا می‌توان مساله کوله‌پشتی را به صورت زیر تعریف کرد:

$$\begin{aligned} \max \quad & imize \quad \sum_{j=1}^n p_j x_j \\ \text{subject} \quad & to \quad C_i : \sum_{j=1}^n w_{ij} x_j \leq b_i \quad , \forall i \in 1..m \\ & x_j \in \{0,1\} \quad , \forall j \in 1..n \end{aligned} \quad (1)$$

² Branch and cut

³ Dynamic programming

⁴ Uni-dimensional

⁵ Iterative

¹ Multiple knapsack problem

۲-۱ اتماتای یادگیر با ساختار متغیر

اتوماتای یادگیر با ساختار متغیر توسط ۵تایی $LA = \{a, b, p, T, c\}$ نشان داده می‌شود که $a = \{a_1, a_2, \dots, a_r\}$ مجموعه عملهای اتماتای یادگیر، $b = \{b_1, b_2, \dots, b_m\}$ مجموعه ورودی‌های اتماتای یادگیر، $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب عملها، $T = p(n+1) = T[a(n), b(n), p(n)]$ الگوریتم یادگیری و $\{c_1, c_2, \dots, c_r\} = \{c\}$ احتمال جریمه شدن هر عمل می‌باشد. اگر در اتماتای یادگیر عمل a_i در مرحله n انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمال‌ها کاهش می‌یابند و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمال‌ها افزایش می‌یابند. تغییرات به گونه‌ای صورت می‌گیرد تا حاصلجمع $(p_i(n))$ ها همواره ثابت و مساوی یک باقی بماند.

الف- پاسخ مطلوب

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (2)$$

ب- پاسخ نامطلوب

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (3)$$

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت مختلف را می‌توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} می‌نامند. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{Rep} و زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} می‌نامند. برای مطالعه بیشتر در رابطه با اتماتاهای یادگیر می‌توان به [۱۲، ۱۳] مراجعه نمود.

۳- حل مساله کوله‌پشتی توسط اتماتاهای یادگیر

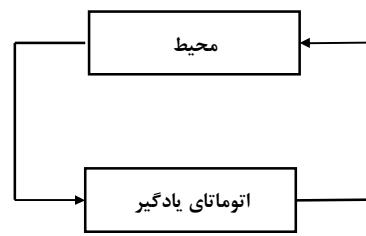
در الگوریتم پیشنهادی که آنرا MKPLA می‌نامیم، مساله با یک گراف کامل مدل می‌شود که هر گره از گراف متناظر با یک کالا در مساله کوله‌پشتی است. هر گره از گراف به یک اتماتای یادگیر با دو عمل انتخاب کالا برای قرار گرفتن در کوله‌پشتی‌ها و عدم انتخاب کالا مجهز است.

برای فعالسازی اتماتاهای یادگیر از عاملها استفاده می‌شود. در هر تکرار از الگوریتم تعدادی عامل وجود دارد که هر عامل یک گره‌های سازد. در ابتدا یک عامل به صورت تصادفی بر روی یکی از گره‌های گراف قرار می‌گیرد و باعث فعالسازی اتماتای یادگیر آن گره می‌شود. هرگاه یک اتماتای یادگیر فعال می‌شود طبق بردار احتمالات انتخاب عمال خود یکی از دو عمل خود را انتخاب می‌کند. در ابتدای الگوریتم احتمال انتخاب عمل حضور کالا در کوله‌پشتی‌ها برابر ۱ در نظر گرفته

تعدادی عامل بر روی گره‌های گراف قرار داده می‌شوند که وظیفه فعالسازی اتماتاهای یادگیر را بر عهده دارند. هر عامل منجر به یک راه حل می‌شود. با توجه به ارزش راه حل بدست آمده از هر تکرار، بردار احتمالات اتماتاهای یادگیر بروز می‌شود. این روند به دفعات تکرار می‌گردد و در خاتمه الگوریتم، بهترین راه حل بدست آمده به عنوان جواب نهایی انتخاب می‌شود. نتایج شبیه‌سازی‌های انجام گرفته کارایی الگوریتم پیشنهادی را در مقایسه با دیگر الگوریتم‌های گزارش شده نشان می‌دهد. به علاوه برای مسائل با اندازه بزرگ الگوریتم پیشنهادی از سرعت همگرایی بالایی برخوردار است. ادامه مقاله بین صورت سازماندهی شده است. در ابتدا در بخش ۲ اتماتاهای یادگیر به صورت اجمالی معرفی می‌گردد. در بخش ۳ الگوریتم پیشنهادی برای حل مساله کوله‌پشتی ارائه می‌شود. در بخش ۴ نتایج شبیه‌سازی‌های انجام گرفته ارائه می‌شود و بخش پایانی مقاله نتیجه‌گیری می‌باشد.

۲- اتماتاهای یادگیر

اتوماتای یادگیر [۱۲] یک مدل انتزاعی است که می‌تواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده توسط محیط تصادفی ارزیابی می‌گردد و پاسخی به اتماتای یادگیر داده می‌شود. اتماتا از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می‌کند. محیط تصادفی را می‌توان با سه‌تایی $E = \{a, b, c\}$ تعریف نمود که $a = \{a_1, a_2, \dots, a_r\}$ مجموعه ورودی‌ها، $b = \{b_1, b_2, \dots, b_m\}$ مجموعه خروجی‌ها و $c = \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالهای جریمه شدن می‌باشد. هرگاه b_i دو مقداری باشد $b_i = 1$ به عنوان جریمه و $b_i = 0$ به عنوان پاداش در نظر گرفته می‌شود. c_i احتمال اینکه عمل a_i نتیجه نامطلوب داشته باشد، می‌باشد. در محیط پایدار مقادیر c_i بدون تغییر باقی می‌مانند، حال آنکه در محیط ناپایدار این مقادیر در طی زمان تغییر می‌کنند. شکل ۱ ارتباط بین اتماتای یادگیر و محیط را نشان می‌دهد.



شکل (۱): ارتباط بین اتماتای یادگیر و محیط

اتوماتاهای یادگیر به دو گروه اتماتای یادگیر با ساختار ثابت و اتماتای یادگیر با ساختار متغیر تقسیم می‌شوند. در ادامه به شرح مختصری درباره اتماتاهای یادگیر با ساختار متغیر که در این مقاله استفاده شده است می‌پردازیم.

اگر ارزش بهترین راه حل تکرار فعلی از ارزش بهترین راه حل بدست آمده تاکنون کمتر باشد اعمال انتخاب شده توسط اتماتاهای یادگیر گره‌های متضاد با کالاهای شرکت کننده در بهترین راه حل پاداش و اعمال انتخاب شده توسط اتماتاهای یادگیر گره‌ها جریمه می‌شوند. برای جریمه عمل انتخابی از فرمول زیر استفاده می‌شود.

$$p(t+1) = p(t) - g \cdot b \cdot p(t) \quad (6)$$

که $g = \frac{1}{f(c) - c_{\max} + 1}$ و b پارامتر جریمه نامیده می‌شود. کل این فرآیند به صورت تکراری و به دفعات انجام می‌شود تا زمانی که راه حل بهینه حاصل شود یا شرط مازکریم تعداد تکرارهای الگوریتم برقرار شود. شبه کد الگوریتم MKPLA در شکل ۳ نشان داده شده است. هر مقاله باید شامل این بخش‌های اصلی باشد: چکیده، کلمات کلیدی، مقدمه، مطالب اصلی، نتیجه، و مراجع. سایر بخشها مثل سپاسگزاری، ضمایم، و زیرنویس‌ها اختیاری است. این بخشها باید در آخر مقاله و قبل از مراجع قرار گیرند، بجز بخش زیرنویس‌ها که پس از مراجع آورده می‌شود.

۴ - نتایج شبیه‌سازی‌های انجام شده

برای ارزیابی کارایی الگوریتم MKPLA از مجموعه تست پیشنهادی Beasley و Chu [۱۵، ۱۶] که در OR-library [۱۷] موجود است، استفاده شده است. نتایج حاصل از الگوریتم پیشنهادی با چهار الگوریتم دیگر که در [۱۱] گزارش شده‌اند، مقایسه شده است. جزئیات این الگوریتمها در مراجع [۱۰، ۱۱، ۱۶ و ۱۸] آرائه شده است.

در آزمایشات انجام گرفته مقدار پارامتر تصمیم‌گیری q برابر $\frac{1}{3}$ ، مقدار پارامتر پاداش a برابر ۱، مقدار پارامتر a برابر ۱ و مقدار پارامتر β برابر ۳ در نظر گرفته شده است. به علاوه تعداد عاملها در هر تکرار برابر ۳۰ انتخاب شده است. مقدار پارامتر جریمه b در ابتدا ۱ در نظر گرفته شده است اما اگر بهترین نتیجه حاصل شده در ۵۰ تکرار مجدد حاصل شد مقدار b به میزان $\frac{1}{3}$ کاهش می‌یابد. جدول ۱ نتایج حاصل برای ۲۰ نمونه تست با ۱۰۰ کالا و ۵ کوله‌پشتی را نشان می‌دهد. ستون اول جدول شماره مساله در مجموعه تست، ستون دوم جدول بهترین جواب شناخته شده برای مساله که توسط الگوریتم ژنتیک حاصل شده است [۱۵] و ستونهای بعدی نتایج حاصل از اجرای الگوریتم‌های مذکور را نشان می‌دهد. نتایج حاصل از الگوریتم MKPLA شامل میانگین ۲۰ بار اجرای الگوریتم با ۱۰۰۰ تکرار در هر اجرا و متوسط تعداد تکرارهای لازم برای رسیدن به بهترین جواب C^* می‌باشد. همانطور که از جدول ۱ مشخص است الگوریتم پیشنهادی جوابهای بهتری نسبت به الگوریتم‌های گزارش شده تولید می‌کند.

شده است تا کلیه کالاهای شناس انتخاب را پیدا کنند. پس از فعال شدن یک اتوماتای یادگیر اگر عمل انتخابی آن اتوماتا، انتخاب کالا برای قرار گرفتن در کوله‌پشتی‌ها باشد، آن کالا را در لیست کالاهای انتخابی قرار می‌دهیم. سپس عامل از بین گره‌هایی که تاکنون پیمایش نشده‌اند یک گره را انتخاب می‌کند. عامل k گره بعدی را بر اساس رابطه ۴ انتخاب می‌کند.

$$p(j) = \frac{h_j}{\sum_{i \in \text{validset}} h_i} \quad , \quad h_j = \frac{P_j}{\sum_{i=1}^m \frac{w_{ij}}{s_i}} \quad (4)$$

که $validset$ مجموعه گره‌هایی است که تاکنون توسط عامل k پیمایش نشده‌اند و انتخاب آنها باعث انحراف از شرط (۱) نمی‌شود و δ ظرفیت باقی‌مانده از کوله‌پشتی η تا این مرحله می‌باشد. در این رابطه عامل گرهی را انتخاب می‌کند که نسبت سود کالای متضاد با آن به مجموع وزن اشغال کننده از ظرفیت باقی‌مانده کوله‌پشتی‌ها مازکریم باشد. در رابطه فوق عامل k با احتمال q گره بعدی دارای بیشترین احتمال را انتخاب می‌کند و با احتمال $1-q$ گره بعدی را با احتمال متضاد با آن گره به صورت تصادفی انتخاب می‌کند. هر گره تنها یکبار توسط هر عامل پیمایش می‌شود. هرگاه مجموعه $vaildset$ یک عامل تهی شد کار آن عامل به پایان می‌رسد و راه حل ایجاد شده توسط آن عامل طبق الگوریتم شکل ۲ بهبود می‌یابد. برای بهبود راه حل طبق [۱۴] مقادیر LP برای کالاهای محاسبه می‌شود. سپس کالاهای به ترتیب نزولی مقادیر LP بررسی می‌شوند. اگر بتوان کالایی را بدون انحراف از شرط (۱) برگزید، آن کالا به لیست کالاهای انتخابی کوله‌پشتی می‌شود.

```
Procedure improve ( $x$ )
    sort  $x$  as  $x^{LP[j]} \geq x^{LP[j+1]}$ 
    for  $j = 1$  to  $n$  do
        if  $x[j] = 0$  then  $x[j] = 1$ ;
        if any  $C_i$  is violated then  $x[j] = 0$ ;
```

شکل (۲): الگوریتم بهبود راه حل ایجاد شده توسط یک عامل

در هر تکرار از الگوریتم، از ارزش بهترین راه حل ایجاد شده توسط عاملها، برای ارزیابی عمل انتخابی اتماتاهای یادگیر استفاده می‌شود. اگر ارزش کالاهای انتخابی بهترین عامل از ارزش بهترین لیست کالاهای بدست آمده تا این مرحله بیشتر باشد، اعمال انتخاب شده توسط اتماتاهای یادگیر بهترین عامل تکرار فعلی از طریق افزایش احتمال انتخاب آنها پاداش داده می‌شود. فرمول یادگیری برای پاداش عمل انتخابی بصورت زیر می‌باشد:

$$p(t+1) = p(t) + q \cdot a \cdot (1 - p(t)) \quad (5)$$

که $q = \frac{|f(c) - c_{\max}|}{f(c)}$ و $f(c)$ مجموع ارزش کالاهای انتخابی است که در تکرار t تولید شده و C_{\max} ارزش بهترین راه حل بدست آمده تا این مرحله است. a پارامتر پاداش نامیده می‌شود.

Algorithm MKPLA

```

construct a complete graph that associates a node to each object;
equip each node with an learning automaton with tow actions {1,2};
set probability of selecting action 1 to 1;
while not maximum number of cycles reached do
    for each agent  $k$ , construct a solution  $Sl_k$  as follows:
        Randomly choose a first object  $O_i$  and active  $LA_i$  in node  $p_i$ ;
        action = the action chosen by  $LA_i$  in node  $p_i$ 
        if action=1 then  $Sl_k = Sl_k \cup \{O_i\}$ ; //select object
        validset= $\{O_i \in \{1, 2, \dots, n\} / O_i \notin Sl_k \text{ & can be selected without violating resource constraint}\}$ 
        while validset $\neq \emptyset$  do
            Rand= a random variable  $\in [0,1]$ 
            if rand $\leq q$  then choose an object  $e$  validset with maximum  $p(i)$ 
            else choose an object  $e$  validset with probability  $ps(i)$ 
            action = the action chosen by  $LA_i$  in node  $p_i$ 
            if action=1 then  $Sl_k = Sl_k \cup \{O_i\}$ ; //select object
            remove from validset every object that violates some resource constraints.
        end while
        improve  $Sl_k$ 
    end for
    update probability vectors of learning automata
end while

```

شکل (۳): الگوریتم MKPLA برای حل مساله کوله پشتی چندبعدی

جدول (۱): مقایسه نتایج الگوریتم MKPLA با دیگر الگوریتم‌های گزارش شده برای ۱۰۰ کالا و ۵ کوله‌پشتی

شماره مساله	بهترین جواب شناخته شده	L. & M. (میانگین)	FIDANOVA (بهترین)	ALAYA (میانگین)	MKPLA (C*) (میانگین)
۱	۲۴۲۷۴	۲۴۲۴۵	۲۴۱۴۵	۲۴۲۴۷	۲۴۲۷۰
۲	۲۳۵۵۱	۲۳۵۲۷	۲۳۵۲۳	۲۳۵۲۹	۲۳۵۲۷.۲۵
۳	۲۳۵۳۴	۲۳۴۶۳	۲۲۸۷۴	۲۳۴۶۲	۲۳۴۹۵.۳
۴	۲۳۹۹۱	۲۳۴۶۹	۲۳۷۵۱	۲۳۹۴۶	۲۳۹۵۹.۹۵
۵	۲۴۶۱۳	۲۴۵۶۳	۲۴۶.۱	۲۴۵۸۷	۲۴۶۰.۲۴.۹
۶	۲۵۵۹۱	۲۵۵۰.۴	۲۵۲۹۳	۲۵۵۱۲	۲۵۵۴۹.۶۵
۷	۲۳۴۱۰	۲۳۲۶۱	۲۲۲۰.۴	۲۳۳۷۱	۲۳۴۱۰
۸	۲۴۲۱۶	۲۴۱۷۳	۲۲۷۶۲	۲۴۱۷۲	۲۴۲۰۰.۲۴۵
۹	۲۴۴۱۱	۲۴۳۲۶	۲۴۲۵۵	۲۴۳۵۶	۲۴۴۰۰.۸۴۵
۱۰	۴۲۷۵۷		۴۲۷۰.۵	۴۲۷۰.۵	۴۲۷۰.۵
۱۱	۴۲۵۴۵		۴۲۴۴۵	۴۲۴۵۶	۴۲۴۵۰.۹۵
۱۲	۴۱۹۶۸		۴۱۵۸۱	۴۱۹۳۴	۴۱۹۲۶۸۵
۱۳	۴۰۵۹۰		۴۰۹۱۱	۴۰۵۶	۴۰۵۰۲۵.۴
۱۴	۴۲۲۱۸		۴۲۰۲۵	۴۲۱۹۴	۴۲۲۰۰.۴۵
۱۵	۴۲۹۲۷		۴۲۶۷۱	۴۲۹۱۱	۴۲۸۹۸.۱۵
۱۶	۴۲۰۰۹		۴۱۷۷۶	۴۱۹۷۷	۴۱۹۸۸.۷
۱۷	۴۰۵۲۰		۴۴۶۷۱	۴۴۹۷۱	۴۰۵۰۰.۷۹۳
۱۸	۴۳۴۴۱		۴۳۱۲۲	۴۳۳۵۶	۴۳۳۸۶.۲۷
۱۹	۴۴۰۵۴		۴۴۴۷۱	۴۴۵۰۶	۴۴۵۱۵.۷

پیشنهادی از سرعت همگرایی به مراتب بالاتری برخوردار است.

۵ - نتیجه گیری

در این مقاله یک الگوریتم تکرارشونده مبتنی بر اتماتاهای یادگیر برای حل مساله کوله‌پشتی چندبعدی پیشنهاد شد. نتایج بدست آمده از آزمایشها نشان داد که الگوریتم پیشنهادی برای حل مساله کوله‌پشتی از کارایی بالایی برخوردار است. نتایج شبیه‌سازی‌های انجام گرفته کارایی الگوریتم پیشنهادی را هم از لحاظ کیفیت جوابهای تولید شده و هم از لحاظ سرعت همگرایی به جواب در مقایسه با تعدادی از الگوریتمهای گزارش شده نشان می‌دهد.

جدول ۲ نتایج حاصل برای ۲۰ نمونه تست با ۱۰۰ کالا و ۱۰ کوله‌پشتی را نشان می‌دهد. همانطور که از جدول مشخص است الگوریتم پیشنهادی در اکثر موارد به جوابهای بهتری دست یافته است.

جدول ۳ نتایج حاصل برای ۵ نمونه تست با ۵۰۰ کالا و ۵ کوله‌پشتی را نشان می‌دهد. بهترین نتایج بدست آمده برای این مجموعه در [۲۰] گزارش شده است. در این مقاله برای حل مساله کوله‌پشتی یک الگوریتم ترکیبی که از جستجوی tabu و برنامه‌نویسی خطی استفاده می‌کند، گزارش شده است. نتایج حاصل از الگوریتم MKPLA شامل میانگین ۱۰ بار اجرای الگوریتم با ۱۰۰۰ تکرار در هر اجرا می‌باشد. همانطور که از جدول مشخص است الگوریتم پیشنهادی در همه جوابهای بهتری نسبت به الگوریتم Alaya تولید کرده است. به علاوه الگوریتم

جدول (۲): مقایسه نتایج الگوریتم MKPLA با دیگر الگوریتمهای گزارش شده برای ۱۰۰ کالا و ۱۰ کوله‌پشتی

شماره مساله	بهترین جواب شناخته شده	L. & M. (میانگین)	ALAYA C*	MKPLA C*	میانگین
۰	۲۳۰۶۴	۲۲۹۹۶	۲۳۰۱۶	۵۳۸	۲۲۰۵۳.۴
۱	۲۲۸۰۱	۲۲۶۷۲	۲۲۷۱۴	۵۷۵	۲۲۷۳۸.۱۳
۲	۲۲۱۳۱	۲۱۹۸۰	۲۲۰۳۴	۵۹۸	۲۲۰۸۶.۳۸
۳	۲۲۷۷۲	۲۲۶۳۱	۲۲۶۳۴	۷۰۰	۲۲۶۱۹.۷۵
۴	۲۲۷۵۱	۲۲۵۷۸	۲۲۵۴۷	۶۴۰	۲۲۵۹۶
۵	۲۲۷۷۷	۲۲۵۶۵	۲۲۶۰۲	۶۴۵	۲۲۶۲۸.۲۳
۶	۲۱۸۷۵	۲۱۷۵۸	۲۱۷۷۷	۵۵۲	۲۱۷۸۸.۸۷
۷	۲۲۶۳۵	۲۲۵۱۹	۲۲۴۵۳	۵۸۶	۲۲۵۹۵.۸
۸	۲۲۵۱۱	۲۲۲۹۲	۲۲۳۵۱	۵۳۴	۲۲۴۰.۸.۴
۹	۲۲۷۰۲	۲۲۵۸۸	۲۲۵۹۱	۵۸۸	۲۲۷۰۲
۱۰	۴۱۳۹۵	۴۱۳۲۹	۴۱۳۲۹	۵۰۱	۴۱۲۶۷۶۵
۱۱	۴۲۲۴۴	۴۲۲۱۴	۴۲۲۱۴	۵۵۹	۴۲۱۶۴.۴۵
۱۲	۴۲۴۰۱	۴۲۳۰۰	۴۲۳۰۰	۵۸۴	۴۲۲۸۴.۲
۱۳	۴۵۶۲۴	۴۵۶۲۱	۴۵۶۲۱	۵۶۲	۴۵۴۱۰.۴
۱۴	۴۱۸۸۴	۴۱۷۳۹	۴۱۷۳۹	۵۳۶	۴۱۷۵۷۶
۱۵	۴۲۹۹۵	۴۲۹۰۹	۴۲۹۰۹	۵۲۵	۴۲۹۳۸.۲
۱۶	۴۳۵۵۹	۴۳۴۶۴	۴۳۴۶۴	۵۹۷	۴۳۵۲۲.۷۳
۱۷	۴۲۹۷۰	۴۲۹۰۳	۴۲۹۰۳	۴۳۹	۴۲۸۷۲.۴
۱۸	۴۲۲۱۲	۴۲۱۴۶	۴۲۱۴۶	۵۹۸	۴۲۱۴۷.۹۵
۱۹	۴۱۲۰۷	۴۱۰۶۷	۴۱۰۶۷	۵۴۸	۴۱۱۰۳.۴۷

جدول (۳): مقایسه نتایج الگوریتم MKPLA با دیگر الگوریتمهای گزارش شده برای ۵۰۰ کالا و ۵ کوله‌پشتی

شماره مساله	بهترین جواب شناخته شده	ALAYA	MKPLA	میانگین	C*
۰	۱۲۰۱۳۴	۱۱۹۶۵۸	۸۸۵	۱۱۹۹۹۲	۲۰۲
۱	۱۱۷۸۶۴	۱۱۷۴۲۲	۸۵۷	۱۱۷۶۹۰	۳
۲	۱۲۱۱۱۲	۱۲۰۶۲۲	۸۶۰	۱۲۰۹۷۶	۲۱۲
۳	۱۲۰۸۰۴	۱۲۰۲۷۹	۸۱۴	۱۲۰۶۵۰.۷	۱۲۲
۴	۱۲۲۳۱۹	۱۲۱۸۲۹	۸۲۶	۱۲۲۱۳۳	۲

- مراجع ۷

^۱ Linear Reward Penalty^۲ Linear Reward Epsilon Penalty^۳ Linear Reward Inaction

- [1] Balas, E., and Zemel, E., *An algorithm for large zero-one knapsack problems*, Operation Research. vol. 28, 1130–1154, 1980.
- [2] Plateau, G., and Elkhiel, M., *A hybrid algorithm for the 0-1 knapsack problem*, Methods of Operation Research. vol. 49, pp. 277–293, 1985.
- [3] Pisinger, D., *An exact algorithm for large multiple knapsack problem*. European Journal of Operation Research, vol. 114, pp. 528–541, 1999.
- [4] Pisinger, D., *Algorithms for knapsack problems*. Ph.D. Thesis, DIKU, University of Copenhagen, Report 95/1995.
- [5] Lee, J., Shragowitz, E., and Sahni, S., *A hypercube algorithm for the 0/1 knapsack problem*. Journal of Parallel and Distributed Computing, vol. 5, pp. 438–456, 1988.
- [6] Ibarra, O. H., and Kim, C. E., *Fast approximation algorithms for the knapsack problem*. Journal of ACM, vol. 22, pp. 463–468, 1975.
- [7] Riadl, G. R., *Weight-Codings in a genetic algorithm for the multiconstraint knapsack problem*. Proceedings of the 2002 IEEE Congress on Evolutionary Computation, pp. 1564–1569, 2002.
- [8] Riadl, G. R., *An improved genetic algorithm for the multiconstrained 0-1 knapsack problem*. Proceedings of the 5th IEEE Conference on Evolutionary Computation, Anchorage, Alaska, pp. 207–211, May 1998.
- [9] Khuri, S., Back, T., and Heitkötter, J., *The Zero/One multiple knapsack problem and genetic algorithms*. Proceedings of the 1994 ACM symposium of Applied Computation, pp. 188–193, 1994.
- [10] Fidanova, S., *Evolutionary algorithm for multidimensional knapsack problem*. Proceedings of PSNVII, 2002.
- [11] Alaya, I., Solnon, Ch., and Ghedira, K., *Ant algorithm for the multidimensional knapsack problem*. Dans Proceedings of International Conference on Bioinspired Methods and their Applications, Slovenia, 2004.
- [12] Sutton, R. S., and Barto, A. G., *Reinforcement learning: An introduction*. MA: MIT Press, Cambridge, 1998.
- [13] Thathachar, M. A. L., Sastry, P. S., *Varieties of learning automata: An overview*. IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, 32, 6, 2002.
- [14] Pirkul, H., *A Heuristic solution procedure for the multiconstrained zero-one knapsack problem*. Naval Research Logistics 34, pp. 161–172, 1987.
- [15] Chu, P. C., *A genetic algorithm approach for combinatorial optimization problems*. Ph.D. Thesis at the Management School, Imperial College of Science, London, 1997.
- [16] Chu, P. C., and Beasley, J. E., *A genetic algorithm for the multidimensional knapsack problem*. Journal of Heuristics 4, pp. 63–86, 1998.
- [17] Beasley, J. E., *OR-Library: Distributing test problems by electronic mail*. Journal of Operational Research society 41, 1069–1072, 1990.
- [18] Leguizamon, G., and Michalewicz, Z., *A new version of Ant System for subset problem*. Proceedings of Congress on Evolutionary Computation, pp. 1459–1464, 1999.
- [19] Vasquez, M., and Hao, J. K., *A hybrid approach for the 0-1 multidimensional knapsack problem*. IJCAI-01, Washington, 2001.