به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس یادگیری ماشین

استاد ناظرفرد

تمرین چهارم

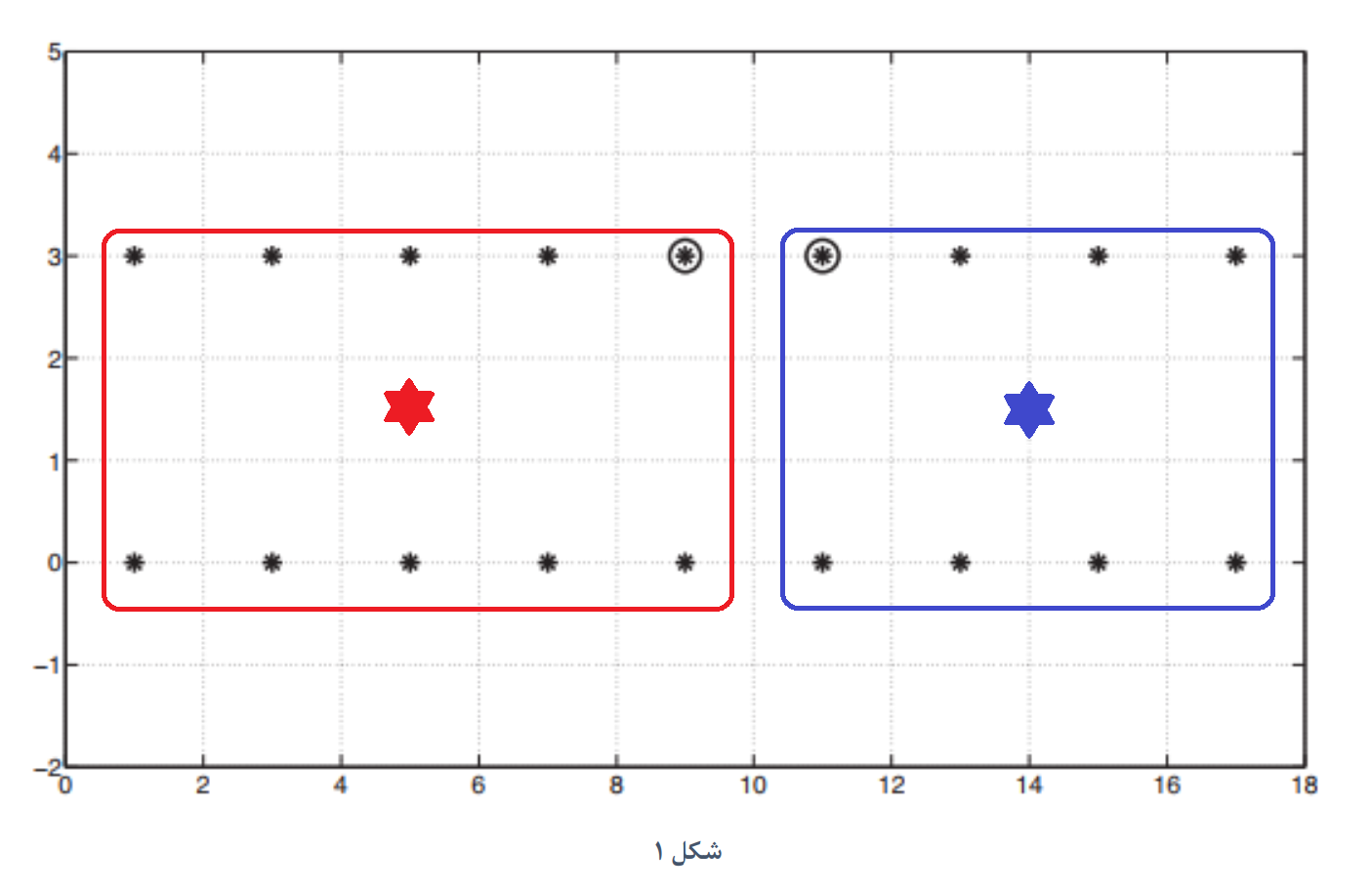
علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

بخش اول: پرسش‌های تشریحی

سوال ۱

دو خوشه آبی و قرمز به همراه مراکز که با ستاره نشان داده شده‌اند در تصویر زیر آورده شده است:



سوال ۲

الف) وقتی که یک الگوریتم خوشه‌بندی را بر روی داده‌های کاربران اعمال کنیم،‌ کاربرانی در یک خوشه قرار خواهند گرفت که ویژگی‌های مشابه به یکدیگر داشته باشند. این همان چیزی است که در کاربرد Customer Segmentation‌ مدنظر است.

ب) وقتی که خوشه‌بندی انجام می‌گیرد و مثلا k خوشه داشته باشیم. اگر خوشه‌بندی سخت باشد، هر داده به یک خوشه تعلق دارد و به مابقی نه. پس می‌توان برای هر داده یک بردار k-بعدی در نظر گرفت که هر بعد متعلق به یک خوشه باشد. هر بعد مقدار یک بگیرد اگه داده به آن خوشه تعلق داشته باشد و در غیر این صورت صفر. اگر خوشه‌بندی نرم باشد، هم طبیعتا برای هر بعد مقدار تعلق داده به آن خوشه نشان داده می‌شود. نهایتا اینکه داده‌ها از فضای اولیه با ابعاد احتمالی بالا به ابعاد k کاهش بعد پیدا می‌کنند.

ج) در تعدادی از الگوریتم‌های خوشه‌بندی مانند DBSCAN داده‌های پرت به طور خودکار شناسایی می‌شوند و طبیعتا می‌توان شناسایی کرد که یک داده، داده غیرعادی و پرت هست یا خیر. در سایر الگوریتم‌ها هم باز می‌توان با داشتن یک معیار فاصله محاسبه کرد که یک داده با هر خوشه چه میزان فاصله دارد؛ طبیعی است که اگر داده با تمام خوشه‌ها فاصله بالایی داشته باشد، یک داده عادی نخواهد بود.

د) مشابه با سوال پیاده‌سازی می‌توان پیکسل‌ها را خوشه‌بندی کرد و بدین ترتیب پیکسل‌های هم خوشه را در یک قطعه قرار دارد. جدای از بحث خوشه‌بندی پیکسل، در هر روشی باید قسمتی از عکس را در یک قطعه قرار داد که نوعی شباهت میان ویژگی‌های درون آن قطعه باشد و این چیزی است که خوشه‌بندی انجام می‌دهد.

سوال ۳

اگر حجم داده‌ها پایین باشد و امکان اجرای چندباره الگوریتم فراهم باشد، می‌توان با رنجی از مقادیر الگوریتم DBSCAN را اجرا کرد و یک سری از شروط را در آن چک کرد؛ مثلا تعداد خوشه‌ در یک بازه معقول و متناسب با کاربرد باشد و یا آنکه اندازه خوشه‌ها نسبت به یکدیگر از یک آستانه‌ای کمتر باشد. بدین ترتیب حالتی که شرایط را داشته باشد مورد پذیرش است. به طور مشابه می‌توان به جای شروط باینری، به هر وضعیت یک امتیاز متناسب با آن نسبت داد و مجموعه‌ای از پارامتر‌ها که بهترین امتیاز داشت را انتخاب کرد.

نهایتا باید توجه داشت که با یک سری بررسی آماری روی داده‌ها نظیر میانگین تراکم می‌توان یک رنج معقول اولیه پیدا کرد.

سوال ۴

در حالت value iterationیک فرآیند تکراری طی می‌شود تا برای هر وضعیت مقدار value‌ یا امتیاز آن محاسبه شود. این فرآیند وقتی متوقف می‌شود که value ها همگرا شوند. پس از اتمام و پیداشدن این مقادیر، نوبت به پیداکردن policy بهینه است. policy بهینه بر اساس جدول مقادیر بهینه بدست می‌آید. لذا این فرآیند یک بار بیشتر انجام نمی‌شود. این درحالی است در policy iteration ابتدا یک policy‌ اولیه درنظر گرفته و سپس به صورت تکرارشونده ابتدا بر اساس policy مقادیر وضعیت‌ها مشخص می‌شوند و سپس بر اساس مقادیر وضعیت، policy بهبود پیدا می‌کند. یعنی در این حالت در هر گام policy و value‌ با هم بروز می‌شوند و زمانی که policy همگرا شود الگوریتم متوقف می‌شود.

سوال ۵

الف) هنگامی که قصد ادغام دو خوشه را در الگوریتم سلسله مراتبی را داشته باشیم بسته به معیار فاصله‌های متفاوتی برای خوشه‌ها حاصل می‌شود:

* در Complete Link بیشترین فاصله میان یک عضو از خوشه اول و یک عضو از خوشه دوم به عنوان فاصله دو خوشه درنظر گرفته می‌شود.
* در Single Link کمترین فاصله میان یک عضو از خوشه اول و یک عضو از خوشه دوم به عنوان فاصله دو خوشه درنظر گرفته می‌شود.
* در Average Link میانگین فاصله تمام جفت داده‌ها که یکی از خوشه اول و دیگری از خوشه دوم باشد محاسبه می‌شود.

از نظر پیچیدگی زمانی برای هر سه حالت لازم است تا فاصله تمام جفت داده‌ها محاسبه شود تا بتوان به ترتیب بیشینه، کمینه و میانگین آن را محاسبه کرد. لذا پیاده‌سازی کلاسیک این سه روش تفاوتی از منظر پیچیدگی زمانی با یکدیگر نخواهند داشت.

از نظر حساسیت به نویز، طبیعتا Average Link باتوجه به حالت میانگین‌گیری که دارد کمترین حساسیت را نسبت به داده‌های نویز دارد. بین دو روش دیگر به طور قطعی نمی‌توان نظر داد ولی می‌توان گفت Single Link‌ حساسیت بیشتری به داده‌های نویز دارد[[1]](#footnote-1). وقتی که داده‌های نویز وجود داشته باشد، این داده‌ها در میان خوشه‌های واقعی قرار می‌گیرند و این امکان را ایجاد می‌کنند که برخی از خوشه‌های واقعی در مراحل اولیه در یک خوشه قرار بگیرند، اما طبیعتا Complete Link با این داده‌ها دچار مشکل نمی‌شود.

ب) در معیار Single Link دو خوشه ۱ و ۲ باهم خوشه می‌شوند و خوشه ۳ و ۴ باهم. چراکه در خوشه ۱ و ۲ داده‌هایی به هم خیلی نزدیک هستند. در معیار Complete Link دو خوشه ۱ و ۳ باهم خوشه می‌شوند و خوشه ۲ و ۴ باهم. چراکه یک چپ‌ترین داده خوشه ۱ از راست‌ترین داده خوشه ۲ فاصله زیادی دارد. در معیار Average Link هم خوشه ۱ و ۳ باهم خوشه می‌شوند و خوشه ۲ و ۴ باهم. چراکه به طور میانگین داده‌های ۱ به داده‌های ۳ نزدیک‌تر است تا ۲.

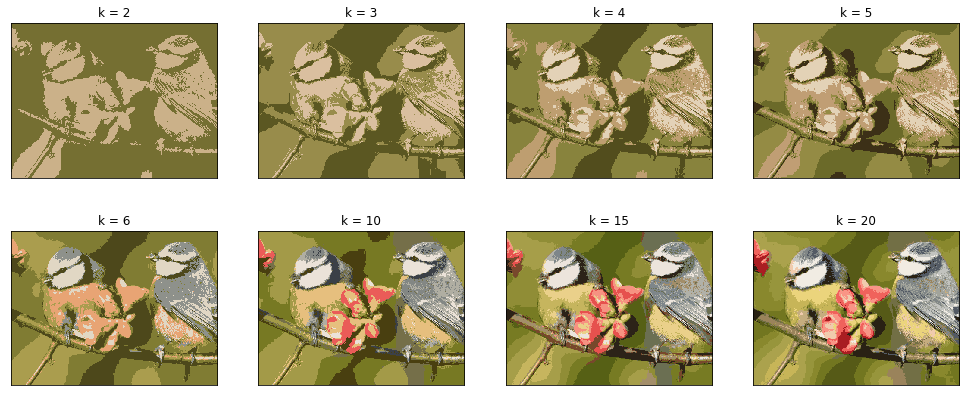
ج) برای مجموعه b معیار Single Link جواب می‌دهد چراکه موقع اجرای الگوریتم و در گام‌های اول که فاصله کم در نظر گرفته می‌شود تمام داده‌های یک خوشه به هم متصل می‌شوند چرا که در این معیار ملاک نزدیک‌ترین داده است و هر زیرخوشه از هر خوشه به زیرخوشه مجاور دیگری از آن خوشه دارای فاصله بسیار کمی است. اما طبیعی است که معیار Complete Link جواب ندهد. چراکه دو لبه‌ی هر خوشه از هم بسیار فاصله دارند و زمانی که قرار است تمام داده‌های یک خوشه به هم متصل شوند این مقدار فاصله برای دو زیرخوشه نهایی هر خوشه وجود خواهد داشت. در Average Link هم مشکل وجود خواهد داشت. چراکه داده‌های دو لبه‌ی هر خوشه میانگین فاصله را بالا می‌برند و این احتمال وجود دارد که لبه‌ی یک خوشه با داده‌های مرکزی خوشه دیگر زودتر تشکیل خوشه دهد.

برای مجموعه c اوضاع دو معیار Complete Link و Average Link متفاوت نخواهد. در این حالت روش Single Link هم به مشکل خواهد خورد. چراکه ممکن است لبه‌ی یک خوشه از طریق داده‌های جدید به مرکز یک خوشه دیگر متصل شود.

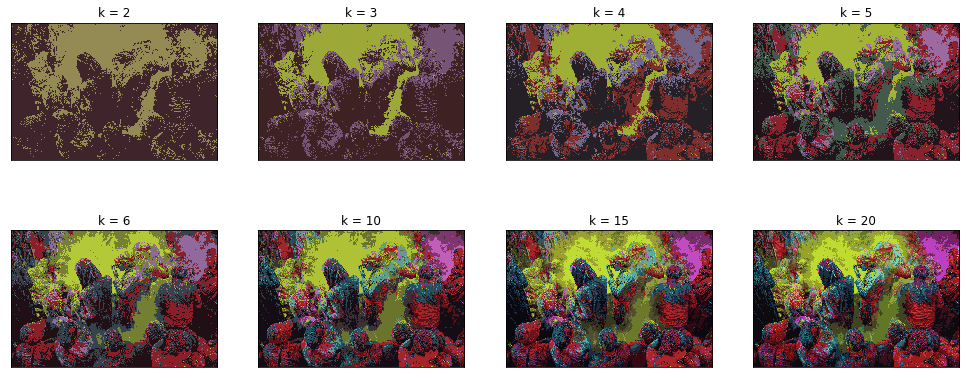
بخش دوم: پیاده‌سازی

سوال ۱

نتایج برای سه تصویر در ادامه آورده شده است:



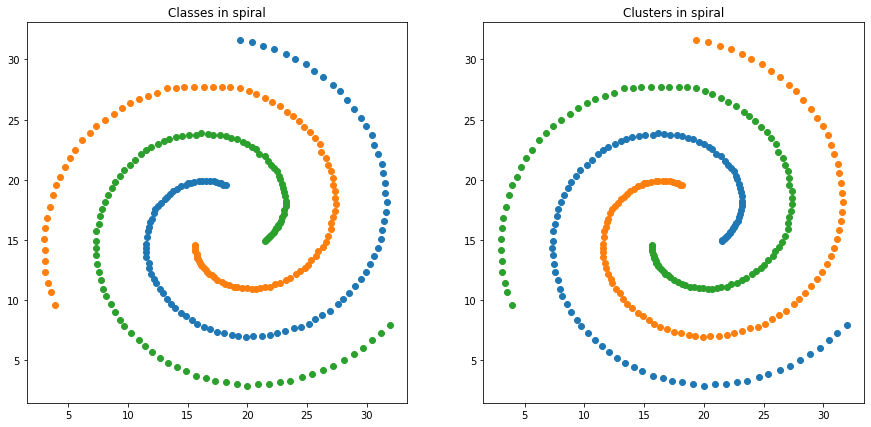




سوال ۲

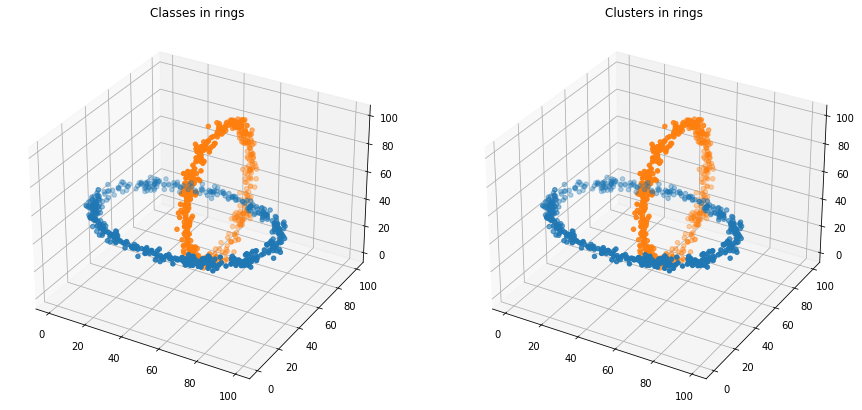
به غیر از مجموعه‌داده rings که دارای سه ویژگی است، مابقی مجموعه‌داده‌ها دارای دو ویژگی هستند. پس از پیاده‌سازی الگوریتم DBSCAN‌ برای هر مجموعه‌داده یک مجموعه پارامتر مناسب که منجر به جواب قابل قبولی شود را بدست آوردم ولی باتوجه به آنکه در سوال تاکیدی بر یافتن جواب بهینه نبود این امکان وجود دارد که جوابی با خروجی بهتر وجود داشته باشد.

**مجموعه‌داده Spiral**



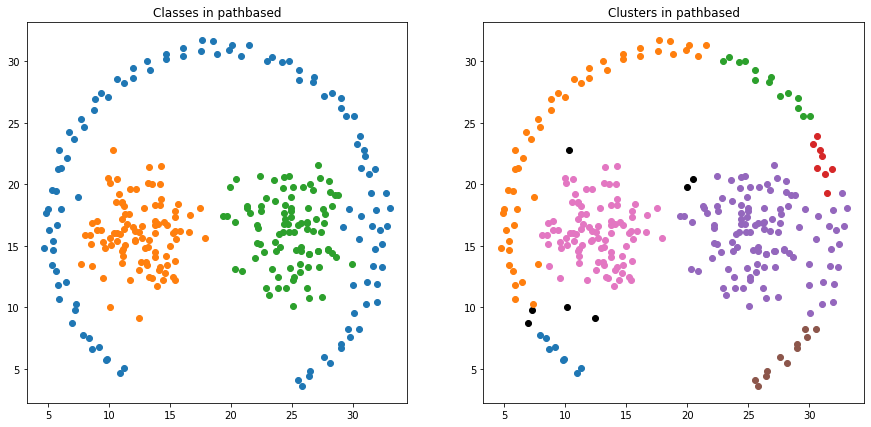
برای این مجموعه‌داده به خلوص ۱۰۰٪ می‌رسیم و در عین حال تعداد خوشه‌های برابر با تعداد خوشه واقعی و عدم تشخیص اشتباه داده پرت را هم داریم. در این شرایط خروجی به بهترین نحو با الگوریتم DBSCAN قابل حصول است. چنین خروجی با الگوریتم‌هایی مانند K-means‌ قابل دست‌یابی نیستند. چراکه میانگین هر خوشه تقریبا در مرکز حلقه‌ها قرار می‌گیرد ولی هر خوشه در یک رشته با تراکم بالا و یکسان قرار گرفته است که مناسب یک الگوریتم برپایه چگالی مانند DBSCAN است.

**مجموعه‌داده Rings**



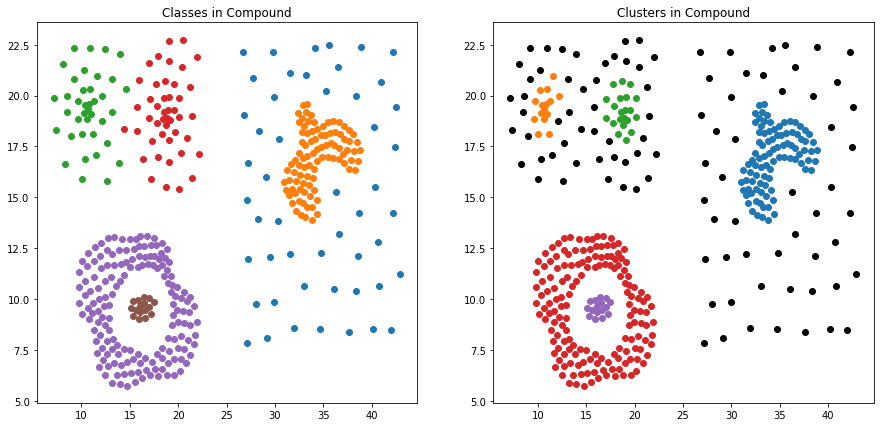
خروجی الگوریتم برای این مجموعه‌داده هم مانند قبل در بهترین حالت ممکن است. از منظر تحلیلی هم مشابه هم هستند.

**مجموعه‌داده Pathbased**



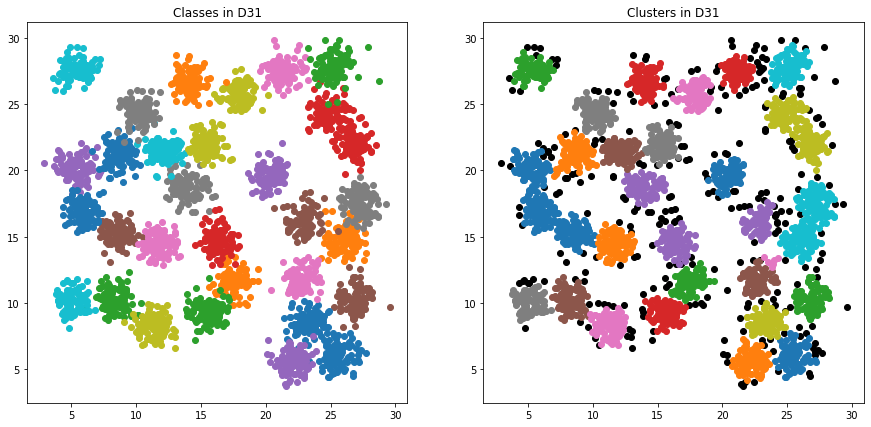
برای این مجموعه‌داده خلوص 92.8٪ بدست می‌آید که مناسب است. اما باید توجه داشت که برای این حالت تعداد خوشه‌ها به جای ۳ به عدد ۷ رسیده است و همچنین حدود ۲٪ داده‌ها نویز شناسایی شده‌اند. خوشه سبز رنگ تقریبا تراکم یکسانی با خوشه حلقوی دارد و طبیعتا کار را برای الگوریتم دشوار کرده است. در عین حال خوشه حلقوی در چند قسمت بریدگی‌هایی دارد که باعث شده تا آن به چندین خوشه بشکند.

**مجموعه‌داده Compound**



برای این مجموعه‌داده خلوص ۱۰۰٪ است. اما نزدیک ۲۵ درصد داده‌ها به عنوان داده‌ی پرت شناسایی شده است. به طوری که یک خوشه کاملا حذف شده است؛ ولی مابقی خوشه‌ها به طور مستقل و یکپارچه شناسایی شده‌اند. مشکلی که الگوریتم برای این مجموعه‌داده با آن موجه است تراکم‌های مختلف خوشه‌ها است. یعنی خوشه قرمز و سبز تراکم کمی دارند ولی خوشه نارنجی و قهوه‌ای تراکم بالایی دارند. در این شرایط یا باید نرخ داده‌های پرت را زیاد کرد و یا آنکه احتمال تجمیع چند خوشه با یکدیگر را پذیرفت که من حالت اول را انتخاب کرده‌ام. نهایتا در مورد خوشه آبی چندان ظاهر یک خوشه را ندارد و انتظار زیادی است که یک الگوریتم خوشه‌بندی بتواند آن را مجزای از بقیه شناسایی کند.

**مجموعه‌داده D31**



خلوص این مجموعه‌داده برابر با 85.7٪، نرخ داده‌های پرت برابر با 9.1٪ است. تعداد خوشه‌های واقعی برابر با ۳۱ ولی تعداد خوشه‌های پیش‌بینی شده ۲۸ تاست. باتوجه به آنکه خوشه‌ها مرز مشترک با یکدیگر دارند،‌ بدیهی است که یا باید نرخ داده‌های پرت را زیاد کرد و یا آنکه احتمال تجمیع خوشه‌های واقعی در یک خوشه را پذیرفت که در اینجا من مجددا مورد اول را انتخاب کردم. به نظر باتوجه به نحوه قرارگیری داده‌ها خروجی قابل قبول است. به علت ادغام ۳ خوشه مقدار خلوص پایین آمده است مگرنه این مولفه هم می‌توانست اعداد بسیاربالایی اتخاذ کند.

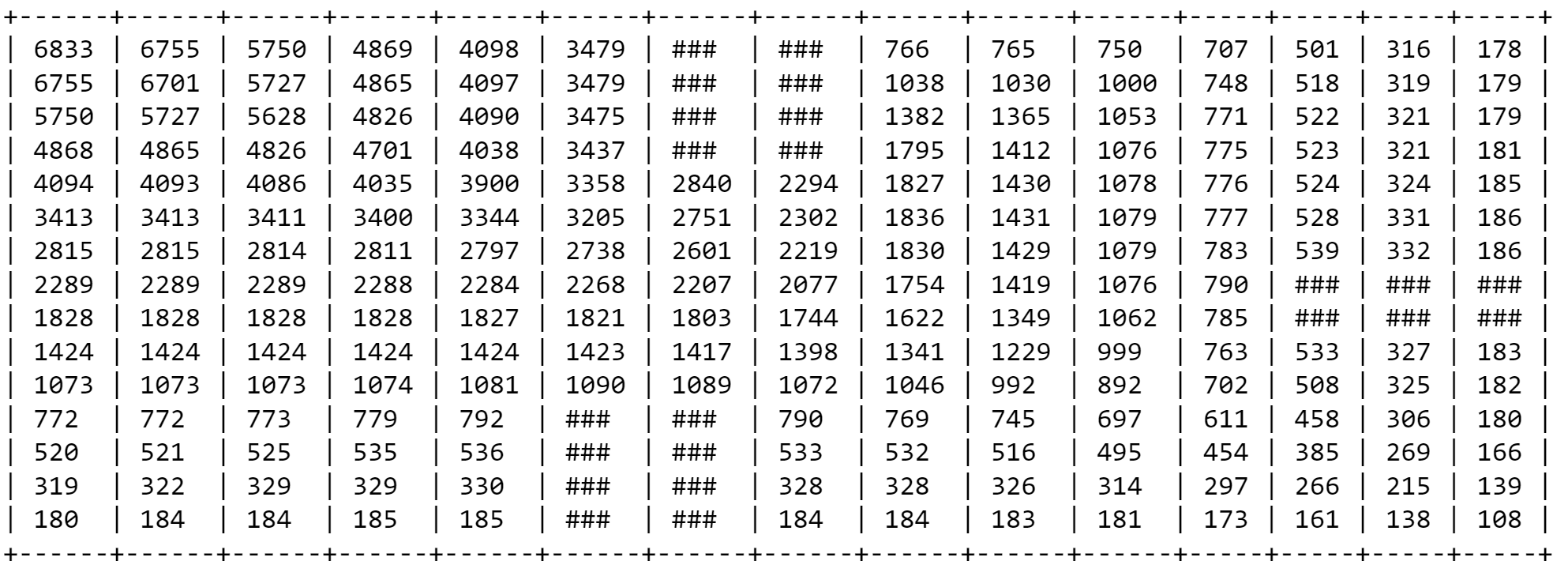
نهایتا توجه کنید که برای این مجموعه‌داده الگوریتم k-means می‌توانست خروجی یکسان و یا بهتر (در صورت تعیین k و مراکز اولیه مناسب) را بدست بیاورد در صورتی که در سایر مجموعه‌داده‌ها چنین چیزی ممکن نیست.

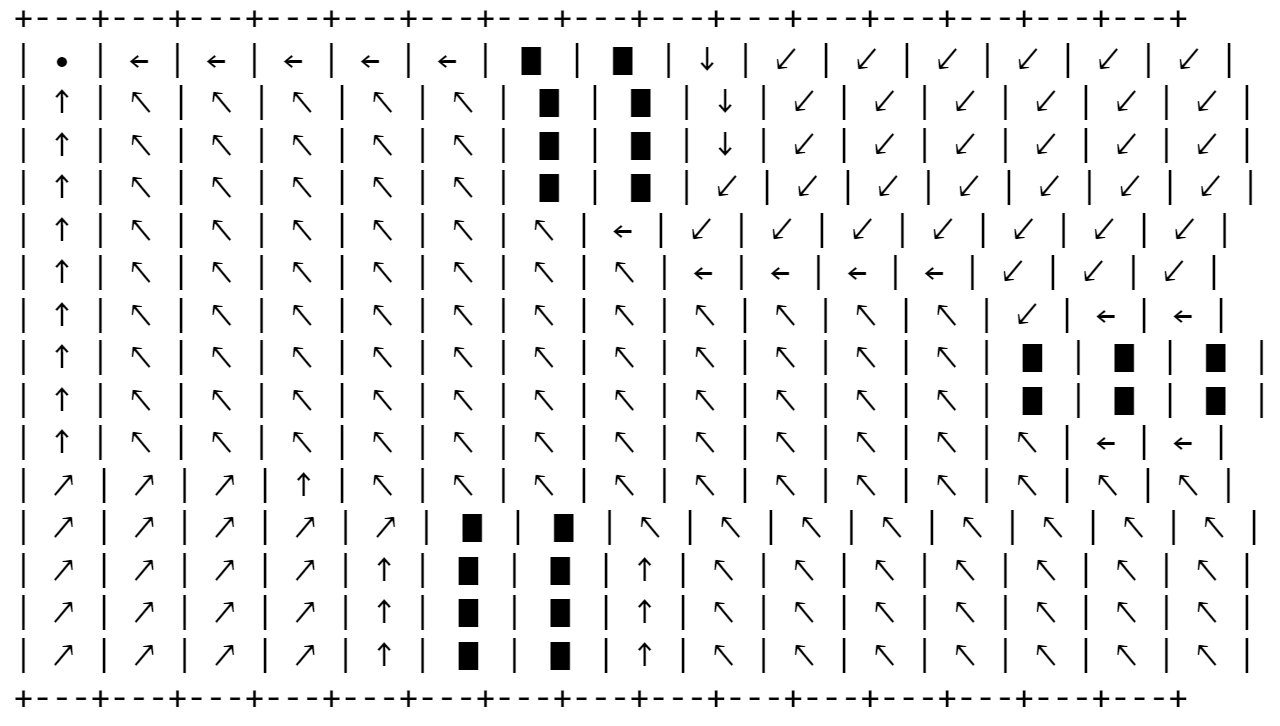
سوال ۳

الف)

سوال ۴

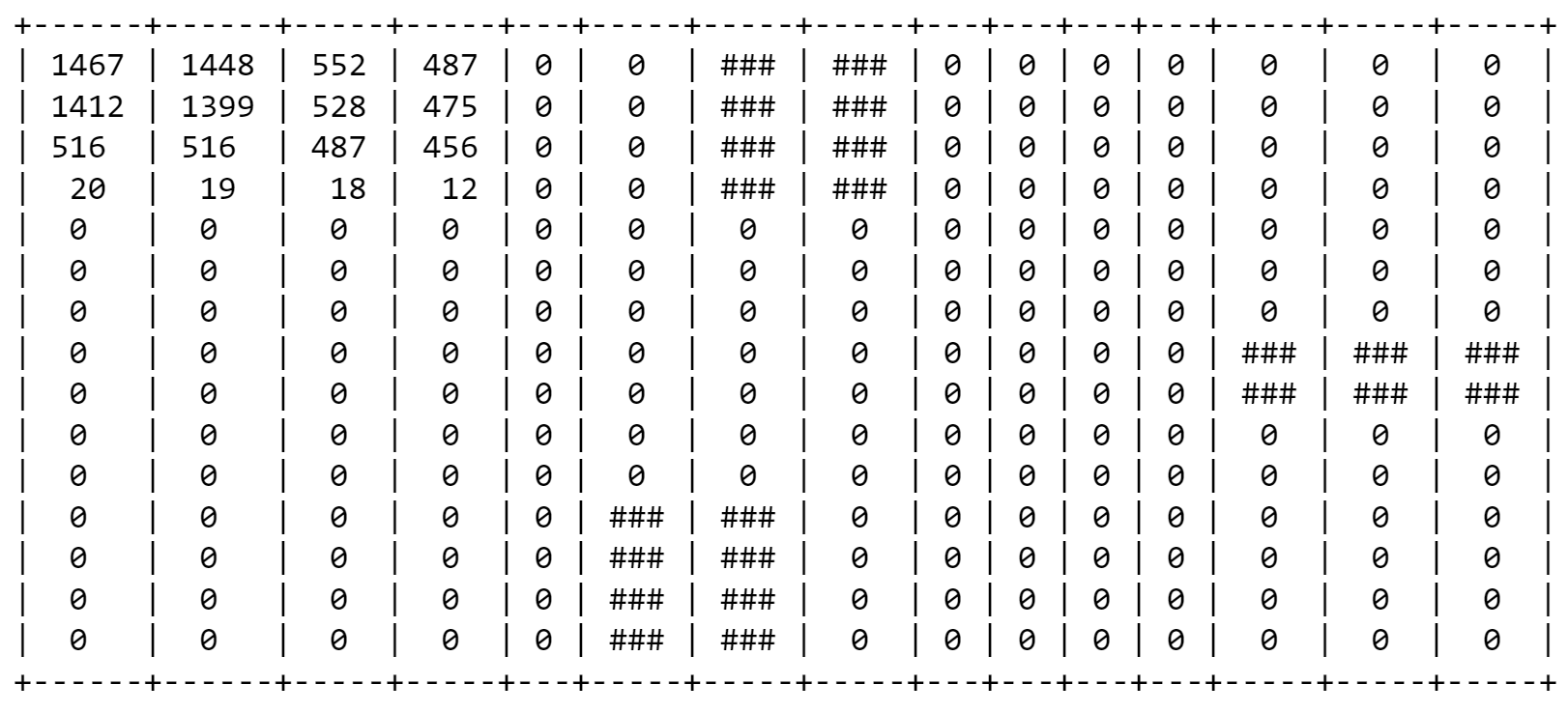
۱) مقدار بهینه و سیاست بهینه به شرح زیر است:

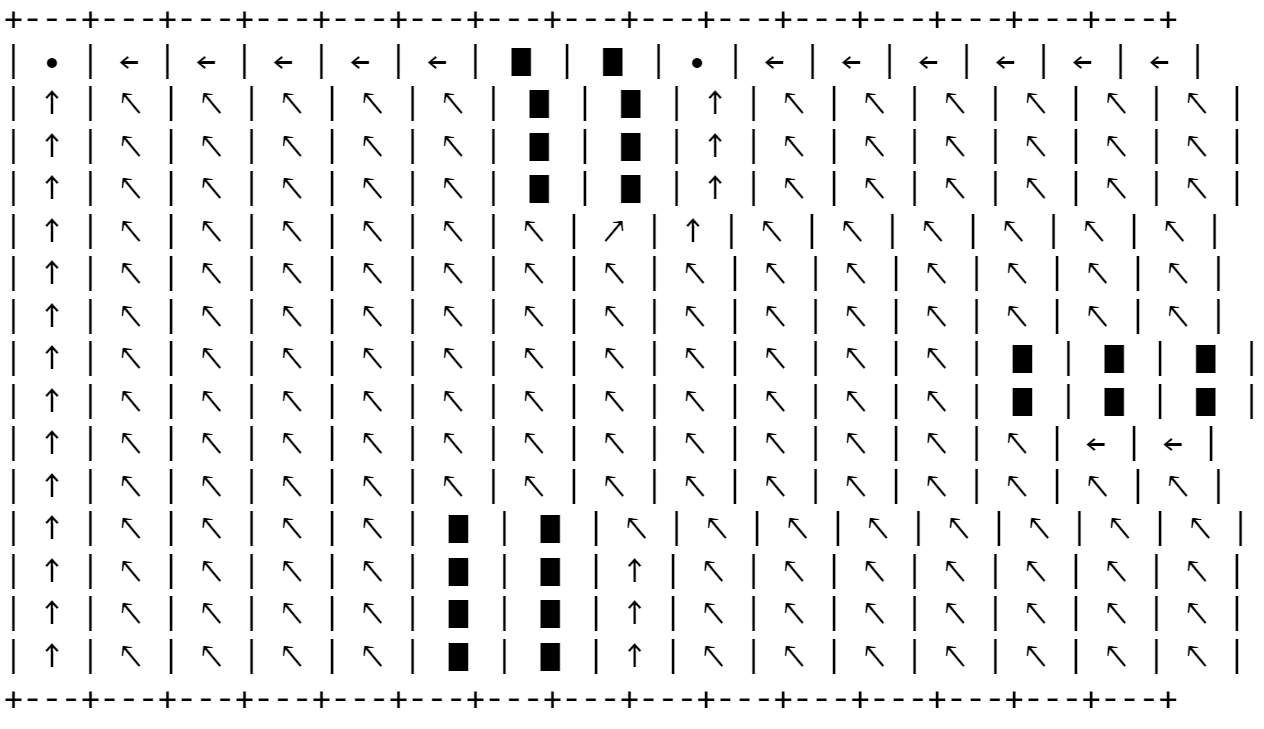




به نظر می‌رسد سیاست‌های اتخاذشده مناسب هستند.

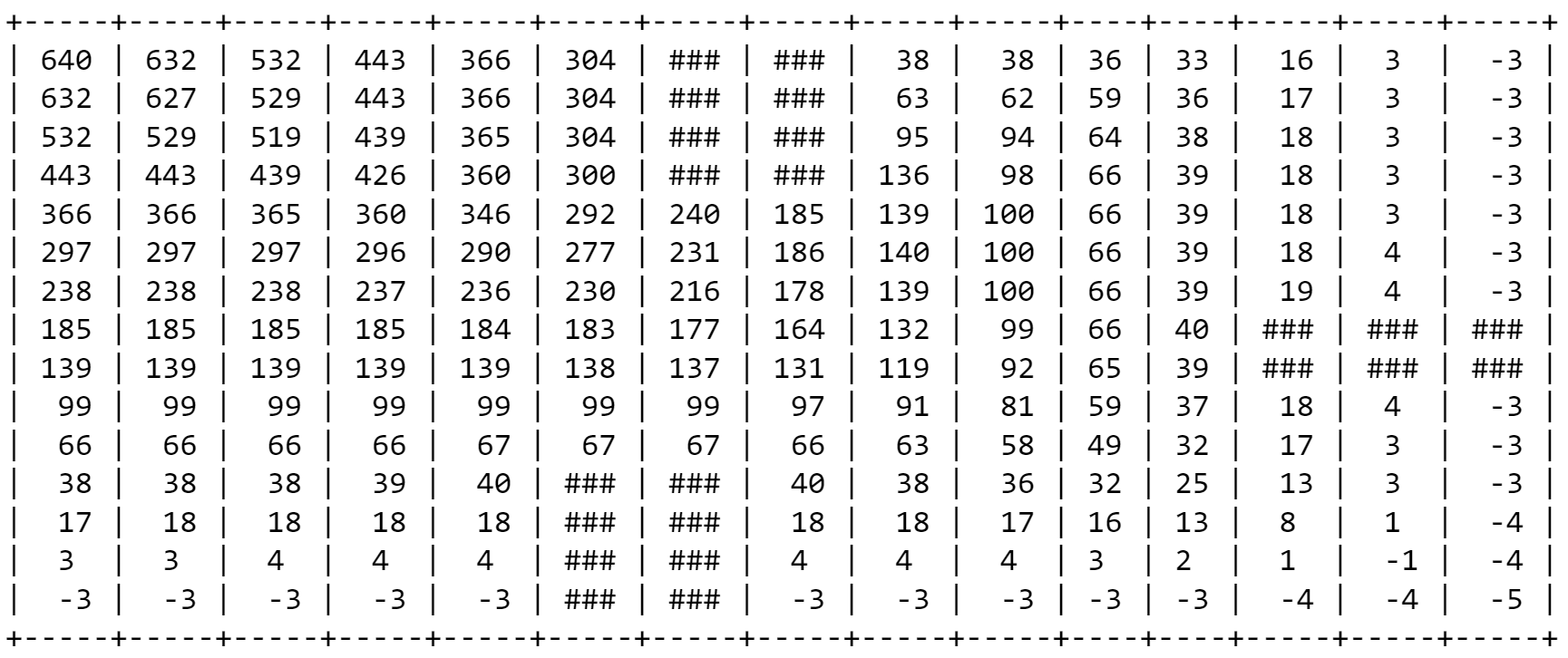
۲) مقدار بهینه و سیاست بهینه به شرح زیر است:

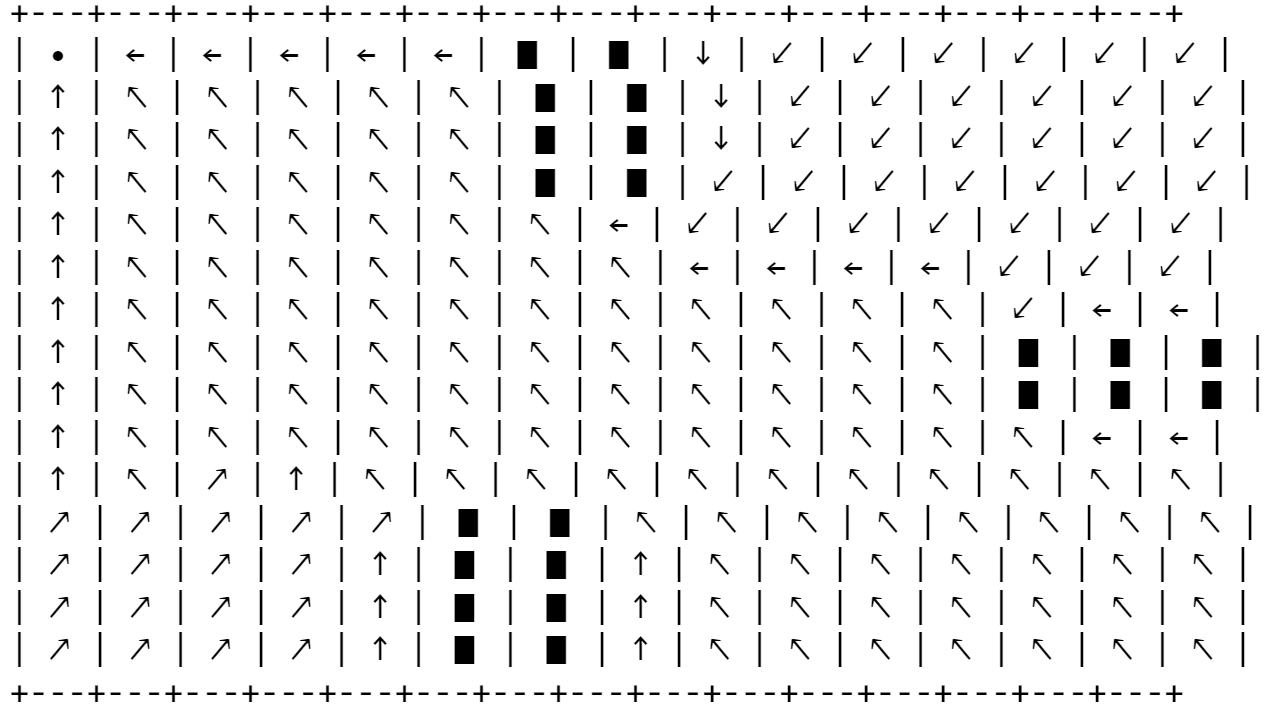




برای این حالت همگرایی سیاست خیلی زود اتفاق می‌افتد و مقادیر وضعیت برای بسیاری از وضعیت‌ها برابر با صفر می‌شود. در این شرایط حرکت بالا چپ به عنوان پیش فرض (آیدی ۰) برای بسیاری از وضعیت‌ها انتخاب می‌شود و لذا به طور اتفاقی برای تعدادی زیادی از وضعیت‌ها پاسخ مطلوب حاصل می‌شود اما با این حال قسمت بالا سمت راست جدول مناسب نیست.

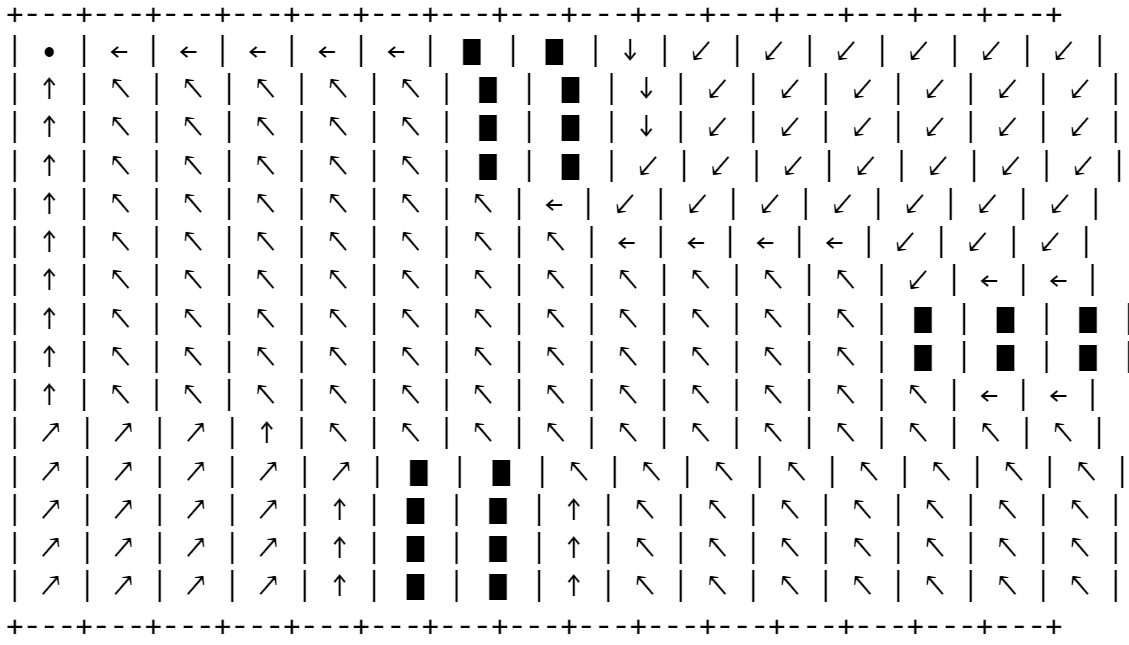
۳) مقدار بهینه و سیاسی بهینه به شرح زیر است:

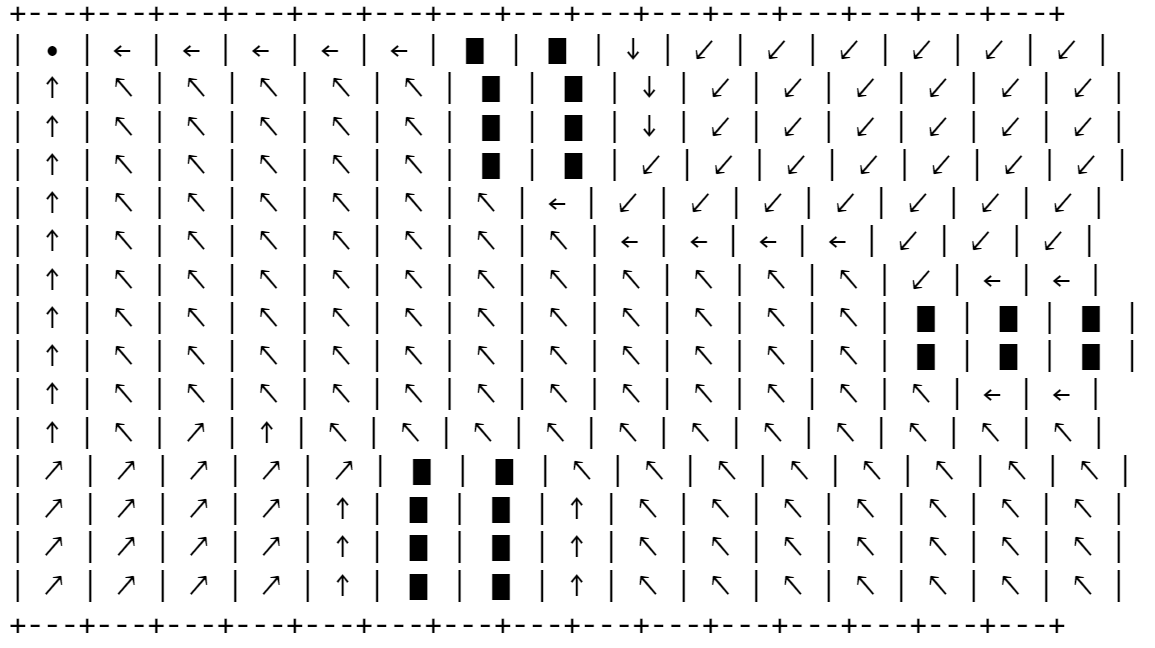


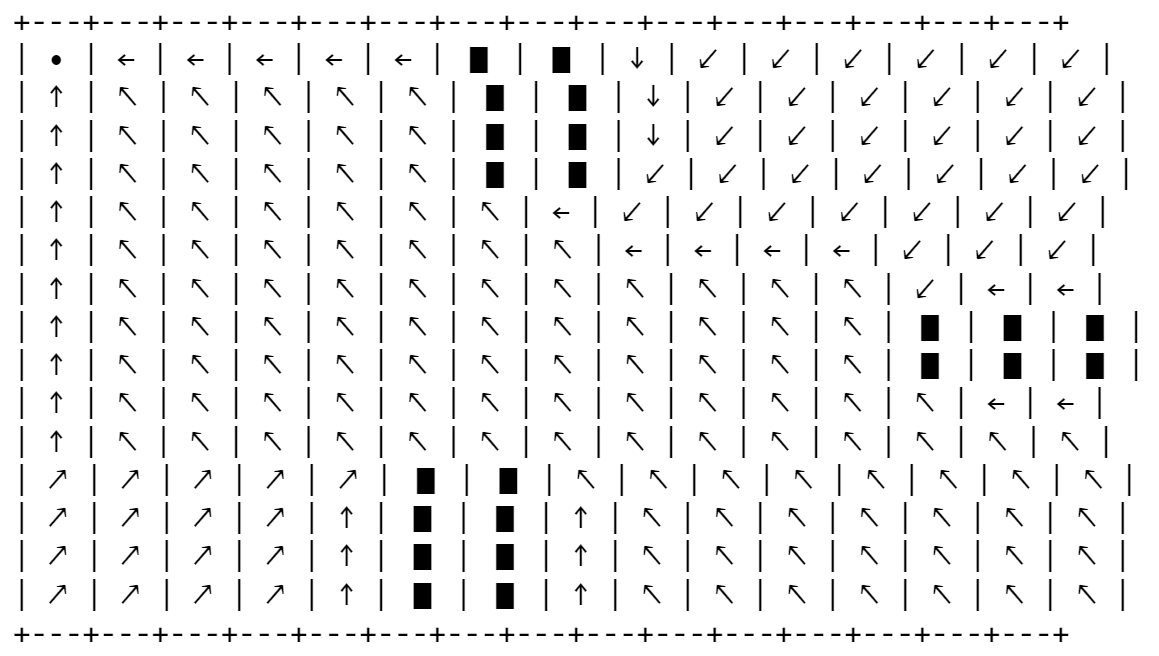


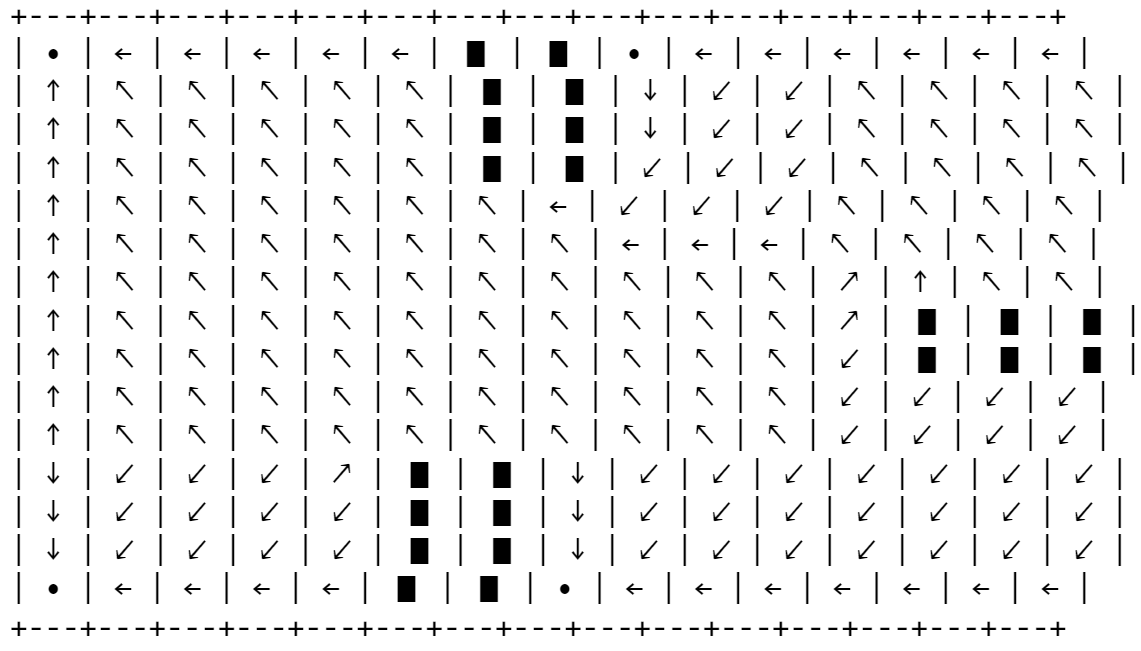
به نظر می‌رسد سیاست‌های اتخاذشده مناسب هستند.

۴) برای تنظیمات قسمت ۱ به ترتیب چهار مقدار ۰.۹، ۰.۷۵، ۰.۵ و ۰.۱ را امتحان کردیم:



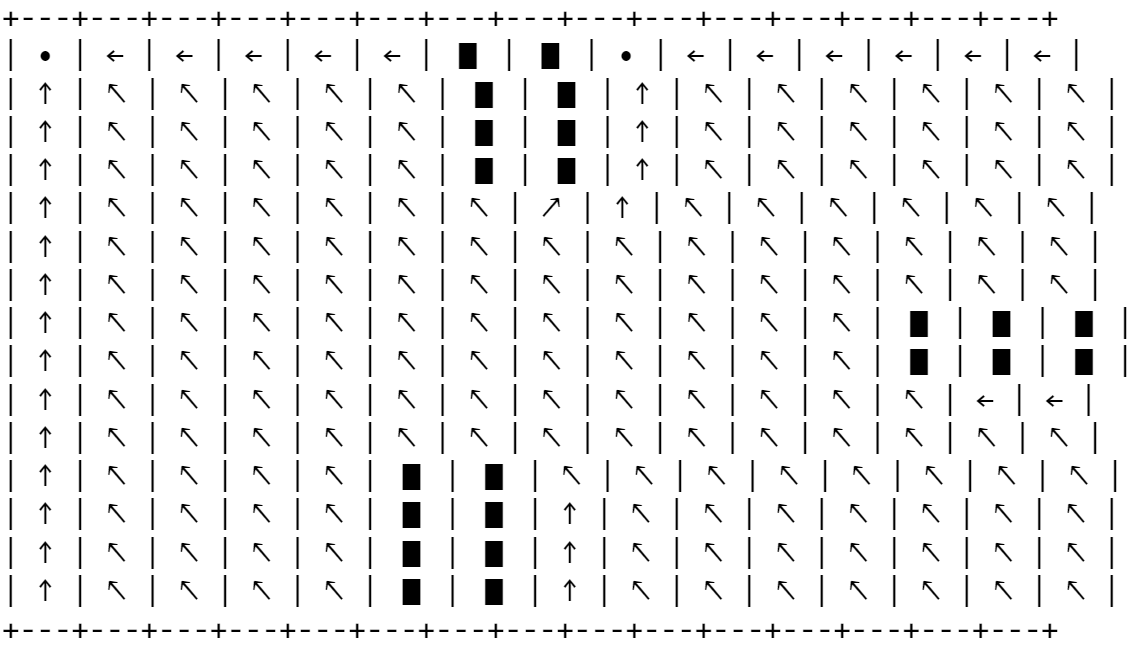


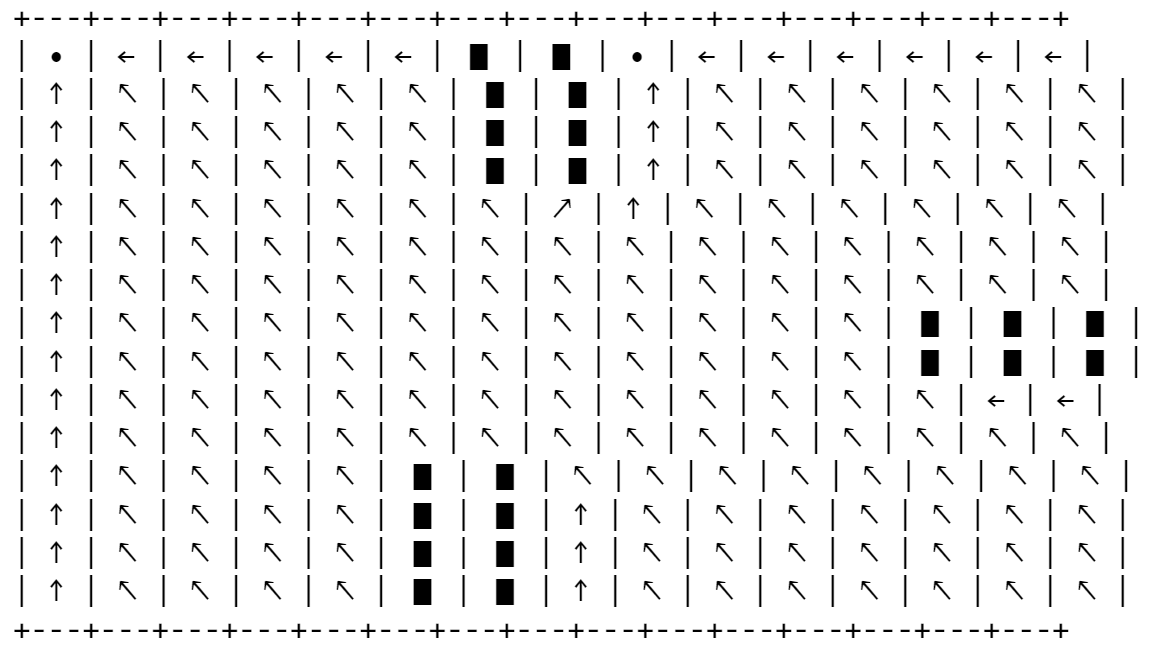


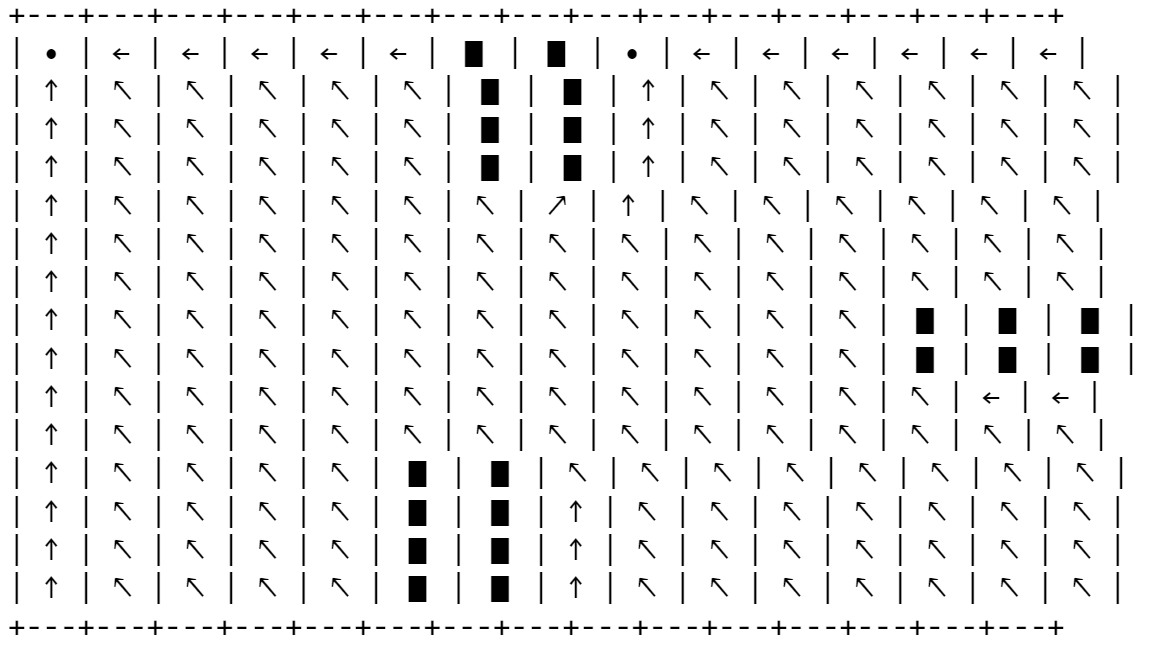


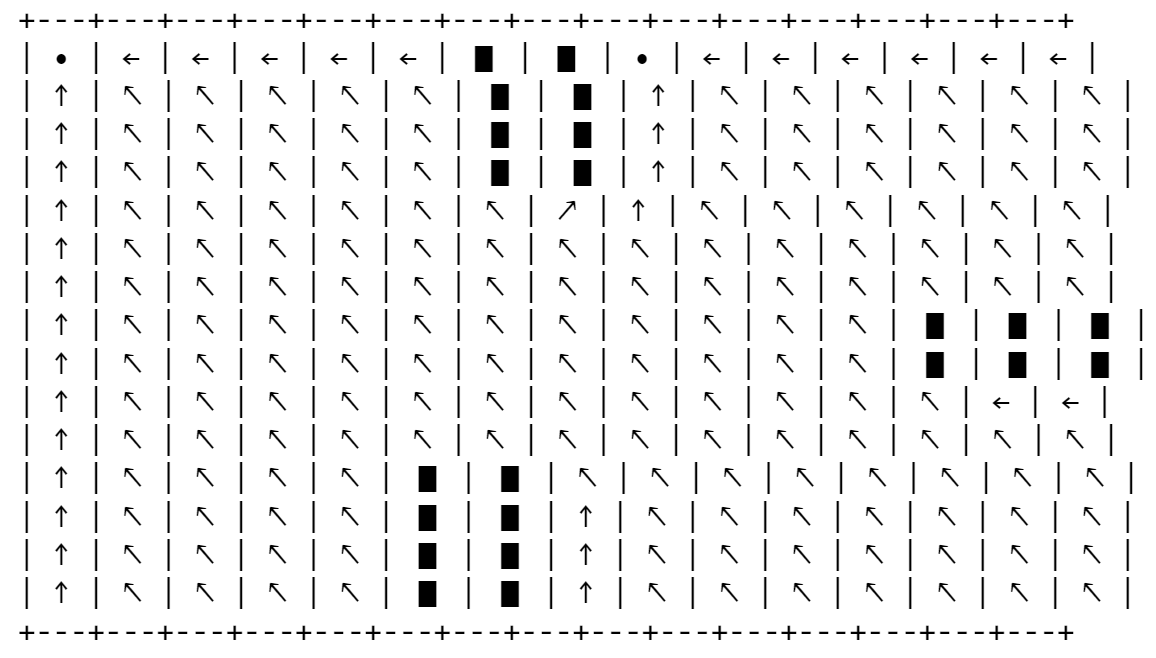
با بررسی حالت‌های مختلف در می‌یابیم که به جز برای مقدار ۰.۱ برای سه حالت دیگر سیاست بهینه یکسان و در یک حالت مناسب قرار دارد. اما برای ۰.۱ برای بخش بالا چپ جدول به نقطه نهایی می‌رسیم و برای قسمت‌های دیگر به وضعیت‌های نامطلوبی خواهیم رفت؛ این نشان می‌دهد که وقتی آینده‌نگری مدل شدیدا کم باشد، مدل نمی‌تواند به جواب مناسبی دست پیدا کند و ترجیح می‌دهد عامل را در نزدیکی جایی که هست نگه دارد.

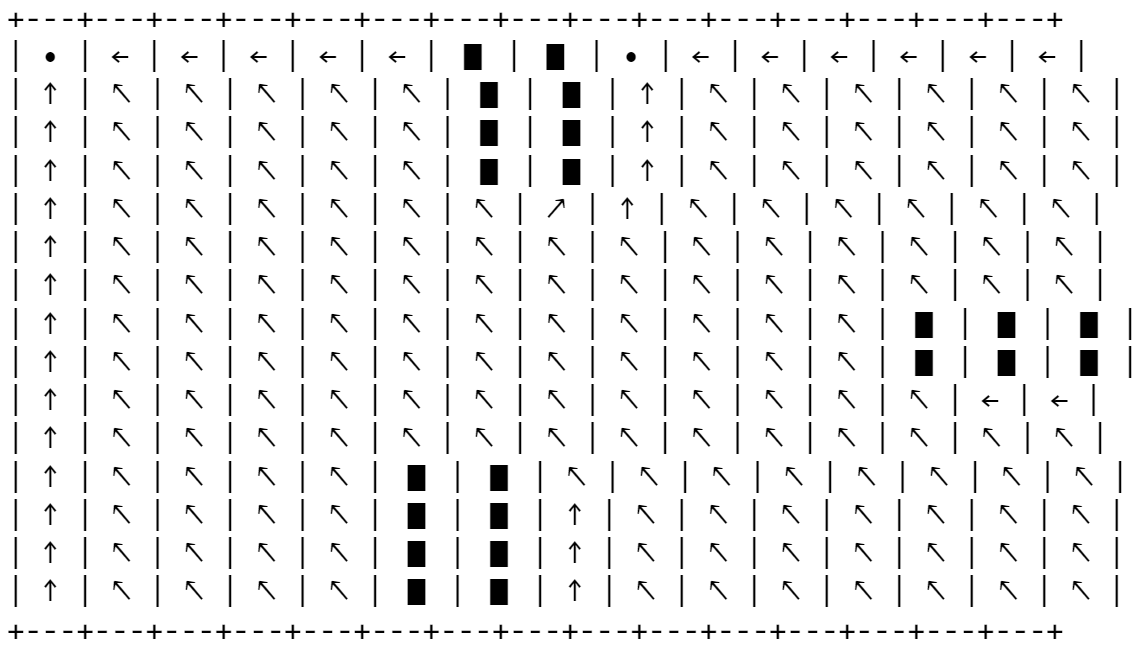
حال چهار حالت discount factor‌ قسمت قبل به علاوه‌ی حالت ۰.۹۹ را برای تنظیم قسمت ۲ بدست می‌آوریم. نتایج سیاست بهینه به ترتیب برای ۰.۹۹، ۰.۹، ۰.۷۵، ۰.۵ و ۰.۱ به شرح زیر است:







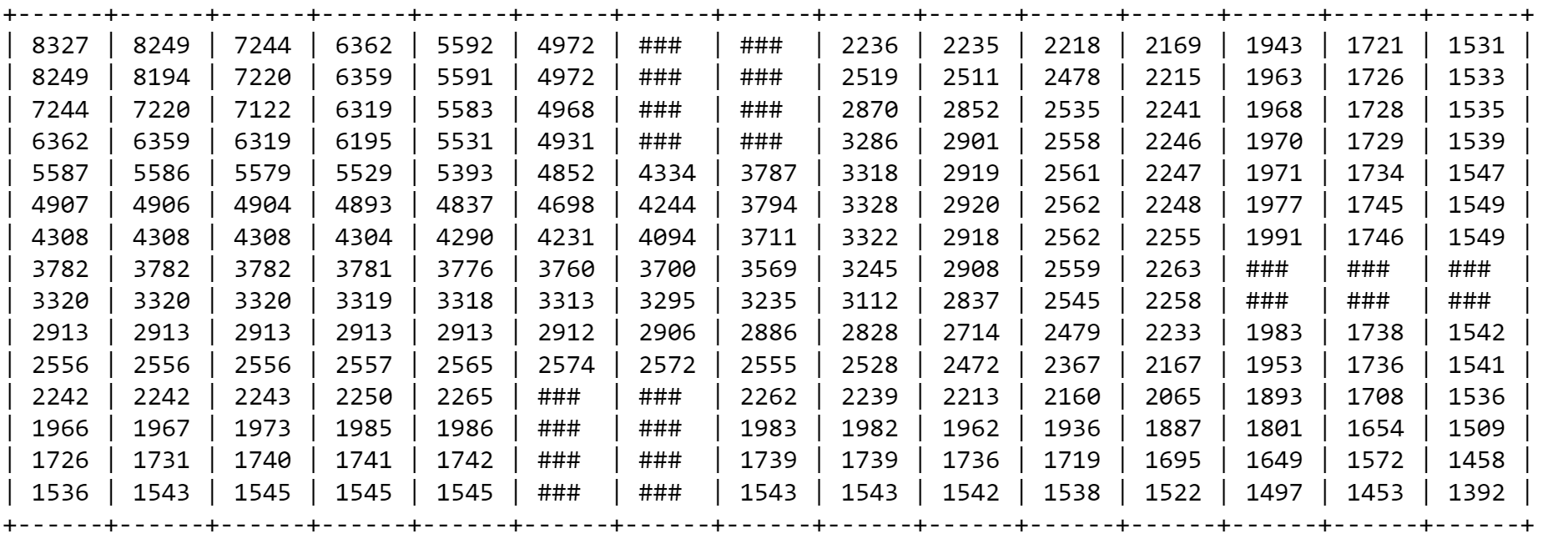


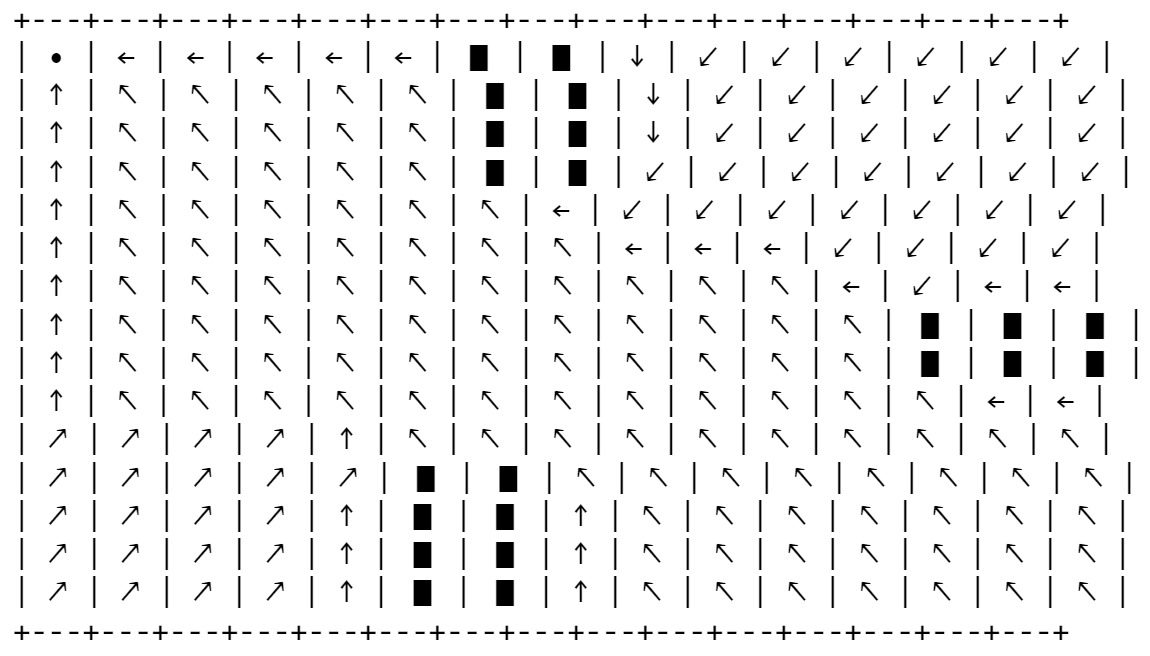


در تمامی این حالات، جدول مقادیر غالبا شامل مقدار صفر است و تنها چندین وضعیت بالای جدول مقدار دارد. با توجه به آنکه حرکت به سمت بالا چپ با آیدی ۰ به نوعی حرکت پیش فرض محسوب می‌شود، این باعث می‌شود تا در ابتدا جدول سیاست در یک وضعیت معقول قرار داشته باشد و همگرایی سیاست در چندین گام اول رخ دهد. اما همانظور که مشخص است برای خانه‌های بالا راست سیاست‌های پیشنهادی مناسب نیستند. تغییر discount factor برای حل این مشکل موثر نیست. شاید تغییر عمل پیش‌فرض بتواند مشکل را حل کند و یا آنکه شاید بتوان پیاده‌سازی را به گونه‌ای تغییر داد که در چند گام اول امکان خاتمه الگوریتم به دلیل همگرایی سیاست وجود نداشته باشد.

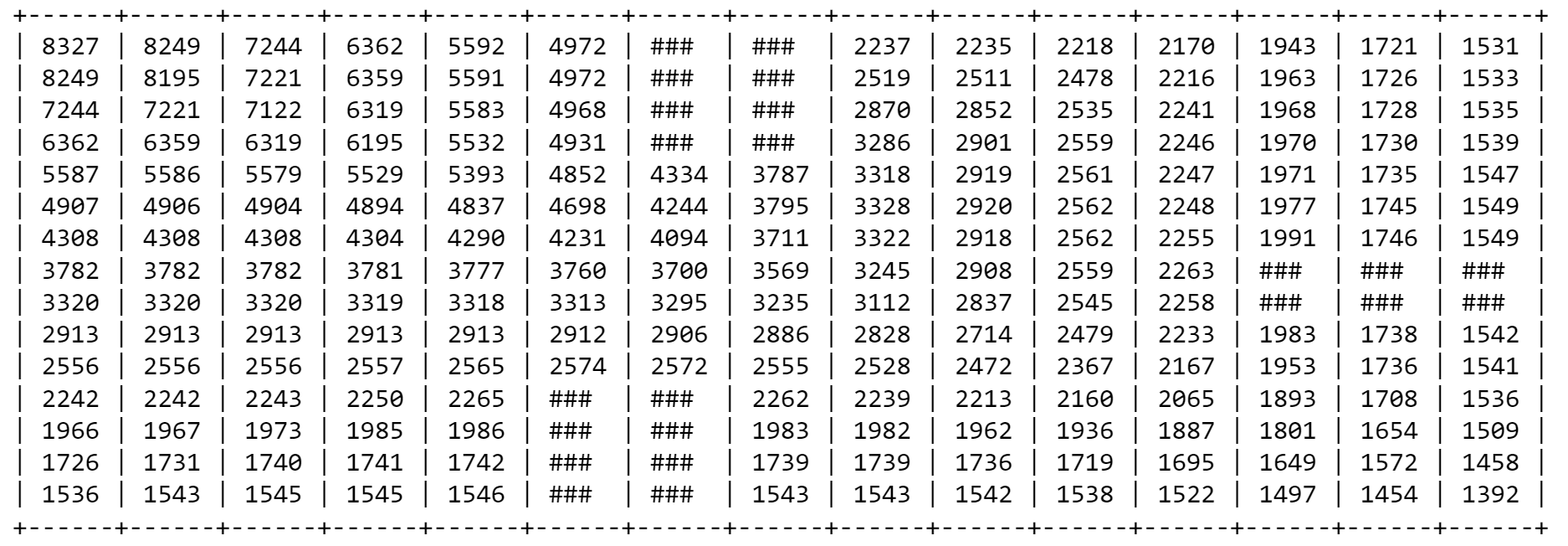
۵) با مقدار discount = 0.9 برای سه محیط پایه، بدون اصطکاک و با اصطکاک زیاد به ترتیب نتایج زیر حاصل شد.

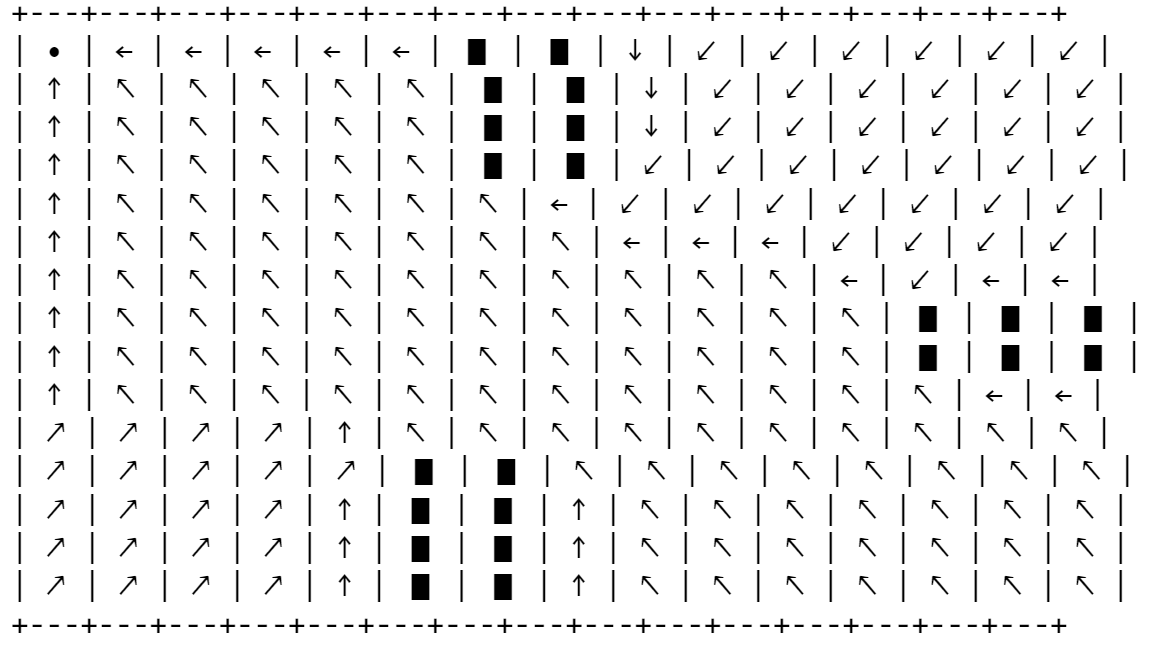
**محیط پایه**



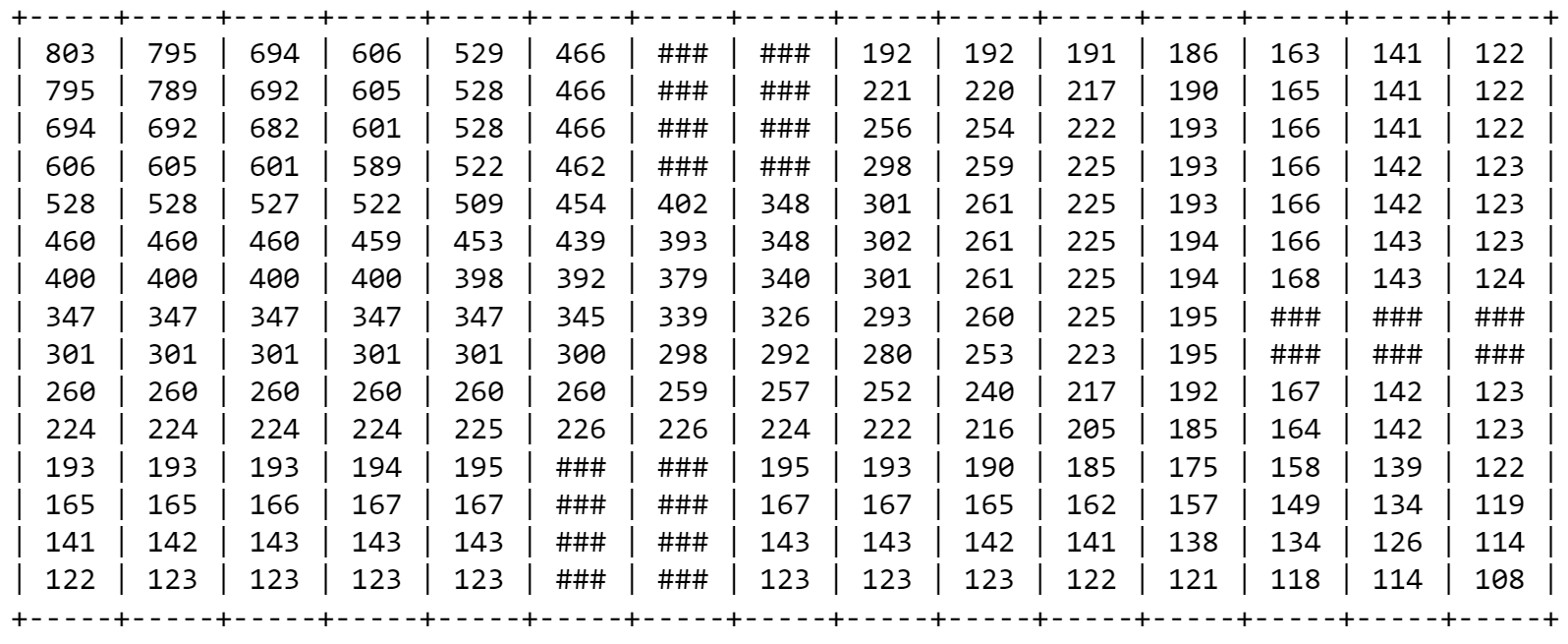


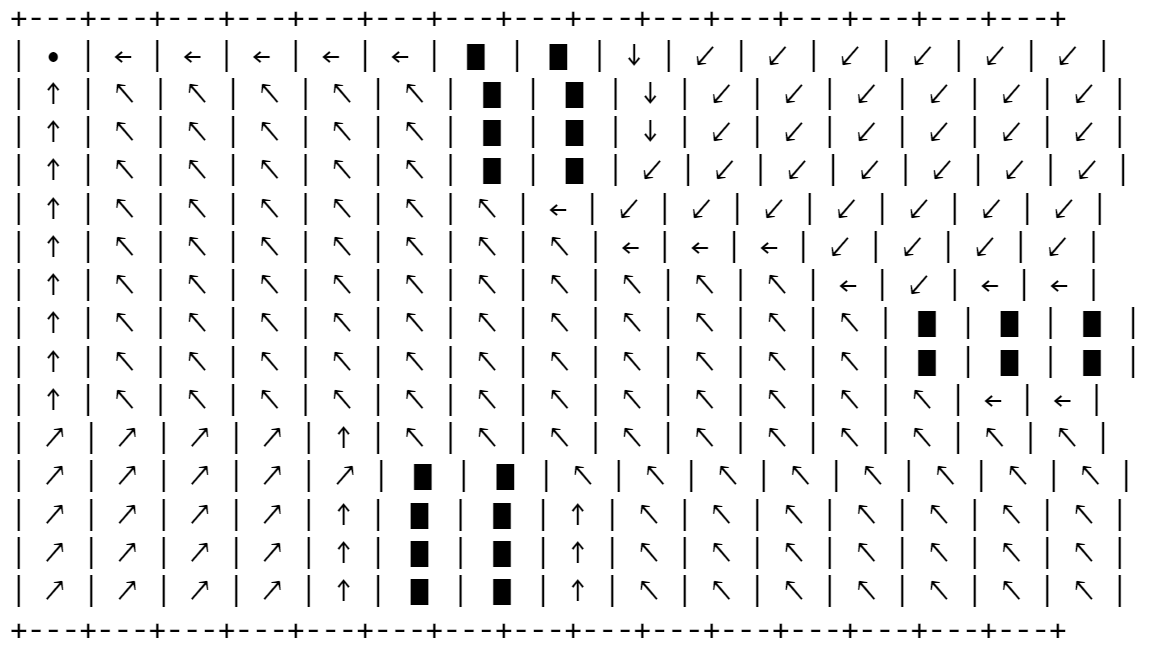
**محیط بدون اصطکاک**





**محیط با اصطکاک زیاد**





در این حالت اجرا، برای هر سه محیط به یک جواب مناسب می‌رسیم این در حالی است که برای محیط بدون اصطکاک در حالت اجرای policy iteration امکان دست‌یابی به جواب مناسب وجود نداشت. در الگوریتم value iteration‌ همگرایی دیرتر رخ می‌دهد و این باعث می‌شود که زمان اجرا نسبت به policy iteration خیلی بیشتر باشد؛ اما از طرفی وقتی همگرایی رخ دهد با اطمینان بیشتری می‌توان الگوریتم را خاتمه داد و امکان دست‌یابی به جواب با کیفیت‌تر بیشتر است.

1. https://stats.stackexchange.com/q/304427/318893 [↑](#footnote-ref-1)